# Adaptive Modelling Against Adversarial Attacks using post-train

## 1. Introduction

Adversarial attacks on deep learning systems pose serious threat to our society. Adding small perturbations to an image that human eyes can hardly perceive could potentially mislead the model to make a completely wrong prediction leading to disasters. Things may got worse when attackers use white-box adversarial attacks as most deep learning models are vulnerable against this kind of attacks. A successful defender model must be able to handle to strongest attack which is white-box as white-box attackers have much more information than black-box attackers.

In this project, we propose a post-training technique to defend against all types of attacks (i.e. including white-box and black-box). Our post-training technique is based on the intuition that an adversarial input is usually near the class decision boundaries because attackers will usually perturb the image just enough to cross the boundary to make the system classify wrongly. Another intuition that we use is that a model post-train with samples from a subset of classes will work more accurately for the chosen subset of classes. Our idea is to ”post-train” the model at inference stage leveraging samples from the original class and the “neighbour” class in the training set. The assumption is that the ”neighbour” class is likely to be the class that will be the goal class of attackers, assuming untargeted attacks (i.e. the attacker does not need to attack the image to fall under a specified class). The model is fine-tuned with small amount of data from these two classes so as to give an accurate prediction. Experiment results showed that our technique can be used to strengthen adversarially trained models.

## 2. Related Work

Deep neural networks are exposed to the risk of adversarial attacks from various known techniques such as the fast gradient sign method (FGSM), projected gradient descent (PGD) attacks, and other attack algorithms. Adversarial training is one of the methods used to defend against adversarial attacks.

Goodfellow(Goodfellow et al., 2014) proposed the fast gradient sign method (FGSM) to generate adversarial samples of clean images using the gradient of the loss function. Later, Goodfellow et al.(Kurakin et al., 2016) enhanced their previous approach by proposing a more advanced algorithm called Basic Iteractive Method (BIM) that was based on the FGSM. BIM applies perturbations to the input image iteratively.

Various methods for defence against adversarial attacks were proposed in the literature. For example, Madry et al. (Madry et al., 2017) suggested using a multi-iteration method called projected gradient descent (PGD) to train the model. They showed that when using the fast-single iteration method (FGSM), the trained model is still vulnerable against stronger multi-iteration adversarial samples. They suggested that by using PGD adversarial training, the model would have better robustness against adversarial samples.

## 3. Methods

The nature of adversarial attack is to find a data point that is near the original input data, but will be classified differently as the original input. The adversarially perturbed data must therefore lie near the class decision boundary as the attacker will also want to minimise the perturbations.

Consider an input data $x$ around the decision boundary between class A and B, as shown in 1 Figure 3. Adversarial attack algorithm will try to generate another input $x'$ that is close to $x$ but on the other side of the boundary. We will call these two input data points the “neighbour” of each other. This neighbour $x'$ of $x$ can be obtained by doing an untargeted adversarial attack on $x$.

Similary, we can also find the neighbour of an adversarial input. Since the adversarial input $x'$ is close to the boundary between classes A and B, the neighbour of $x'$ will likely be in the same class as the natural input $x$ assuming a two class boundary.

In an actual attack and defence scenario, the defender will not know whether an input is natural or adversarial. For any given input $x'$, there are the following three scenarios, as shown in 2:

a) The input is natural without adversarial attack

b) The input is adversarial, but it is not a successful attack

c) The input is adversarial, and it is a successful attack

In all three cases, the true class label comes from either the original class $y'$ or the neighbour class $y''$, as shown in 2. The defender just need to find the original class $y'$ (which is

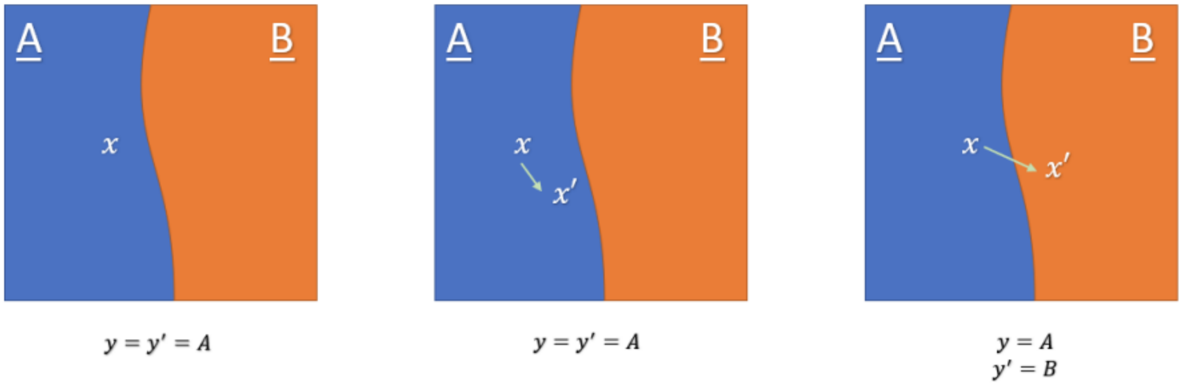

*Figure 1.* Illustration of three different scenarios of input data points around the decision boundary. $x$ is the natural input whereas $x'$ is the adversarial input supplied to the model

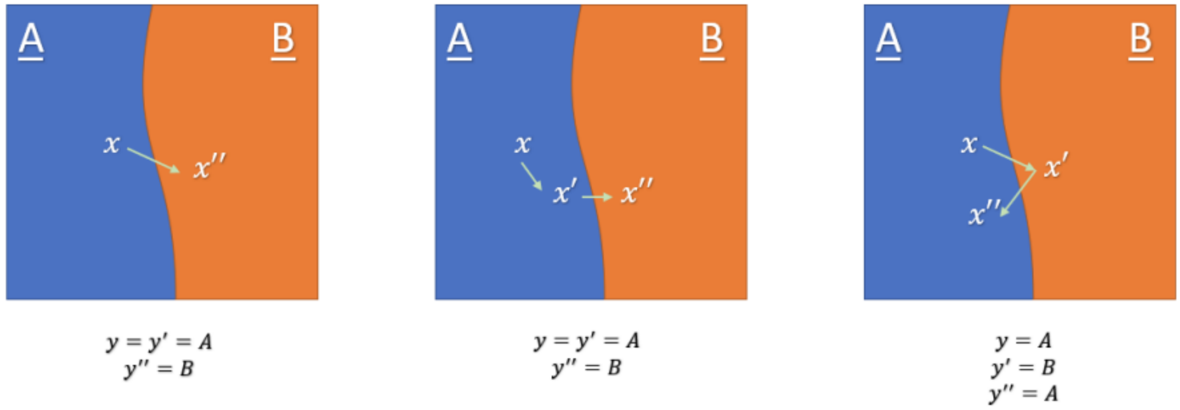

*Figure 2.* Illustration of three different scenarios of input data points around the decision boundary of class A and class B. $x''$ is the neighbour of input $x'$ given to the model.(left) Natural input supplied $x = x'$, then $y = y'$. (middle) Unsuccessful attack attempted, hence $y = y'$. (right) Successful attack occurs, and $y = y''$.

given by the model output) and neighbour class $y''$ (which is given by the model output after the defender's self-inflicted adversarial attack), and post-train with samples from classes $y'$ and $y''$. We would like to highlight that there is no need to know whether an input $x'$ is natural or adversarial.

## 4. Experiments

We tested on Tiny ImageNet dataset. Tiny ImageNet is a subset of the famous ImageNet from Large Scale Visual Recognition Challenge (ILSVRC). Tiny ImageNet contains 100,000 images of 200 classes (500 for each class) downsized to 64×64 colored images.

We used ResNet-50 as our backbone architecture for our study. Our experiment results are shown in the appendix.

## 5. Conclusion

In this report, we described our proposed post-train method to adaptively guard against adversarial attacks. Our method comprises two steps: finding the "neighbour" class and fine-tuning with small amount of data from these two classes to give an output that is more robust agains adversarial attacks. We tested our method on Tiny ImageNet dataset against different adversarial attacks.

## References

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

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

| Train on | using post-train | Normal ACC | Robust ACC |
|---|---|---|---|
| clean | ✓ | 53.8967 | 0.0626 |
| clean | × | 54.4288 | 0.0626 |
| fgsm | ✓ | 36.1567 | 13.5521 |
| fgsm | × | 37.1625 | 12.2287 |
| pgd | ✓ | 29.7818 | 13.5464 |
| pgd | × | 30.5936 | 12.9883 |

*Table 1.* Experiment result of ResNet-50 trained on Tiny-ImageNet with different adversarial attack samples with/without post-train. Normal ACC means Top-1 accuracy testing on original images; Robust ACC means Top-1 accuracy testing on adversarial samples using pgd attack($\epsilon = 8/255, \alpha = 10/255$). Model trained on clean means no adversarial attack samples in training stage;Model trained on fgsm with $\epsilon = 8/255$;Model trained on pgd with $\epsilon = 8/255, \alpha = 3/255$.Noted that $\epsilon$ is perturbation bound, and $\epsilon$ is step size in pgd attack.

## A. Appendix

Github: https://github.com/Yinjie-ZHENG/Adaptive-Modelling-Against-Adversarial-Attacks-using-post-train