TrustWorthy Machine Learning
Assignment N.1

April 24, 2025

Bar, Alon
205476013

---

1 White-box vs. query-based black-box attack

1. Fill out the missing code in utils.compute_accuracy. What is the benign accuracy of the model?

> **Ans:** The model achieves a benign accuracy of 87.50% on the clean test set. This was computed by comparing the model's predictions against ground-truth labels and taking the proportion of correct classifications.
>
> Printed result from the python file:
>
> Benign accuracy: 0.8750

2. Implement attacks.PGDAttack.execute while paying attention to its documentation. It is recommended to use assertions to ensure the adversarial images are valid images and lie within the ϵ -ball centered at their benign counterparts. Add the missing code to utils.run_whitebox_attack and utils.compute_attack_success and run main_a.py. What are the success rates of the untargeted and targeted white-box attacks?

> **Ans:** After running the untargeted and targeted PGD attacks:
>
> - Untargeted attack success rate: 98.00%
> - Targeted attack success rate: 94.50%
>
> Both success rates were measured by the fraction of adversarial examples that fooled the model under each setting.
>
> Printed result from the python file:
>
> White-box attack:
>
> - untargeted success rate: 0.9800
>
> - targeted success rate: 0.9450

3. Fill-out the missing code in attacks.NESBBoxPGDAttack.execute. Notice that the attack will take advantage of the gradients estimated in a black-box manner using NES to produce adversarial examples with PGD. For best performance, we recommend using antithetic sampling when approximating gradients via NES.1 Add the missing code to utils.run_blackbox_attack and execute main_a.py. What are the success rates of the untargeted and targeted query- based black-box attacks with and without momentum? How do they compare to the white-box attacks? Does momentum help improve the success rate or decrease the number of queries? If so, why?

Ans:

The black-box attacks achieve lower success rates compared to the white-box attacks, as expected, because gradient estimates from Natural Evolution Strategies (NES) are inherently noisier and less accurate compared to exact gradients available in white-box scenarios.

Momentum significantly improves the success rate while simultaneously decreasing the median number of queries. This happens because momentum helps smooth out the gradient estimates obtained from NES, thus reducing variance and making the iterative updates more stable and effective.

Targeted	**Momentum**	**Success Rate**	**Num Queries**
Yes	0	0.775	6800
Yes	0.9	0.86	4000
No	0.0	0.93	3600
No	0.9	0.96	2800

Table 1: Black-box Attack Success Rate and Median Number of Queries

Printed result from the python file:

Untargeted black-box attack (momentum=0.00):

- success rate: 0.9300

- median(# queries): 3600

Targeted black-box attack (momentum=0.00):

- success rate: 0.7750

- median(# queries): 6800

Untargeted black-box attack (momentum=0.90):

- success rate: 0.9600

- median(# queries): 2800

Targeted black-box attack (momentum=0.90):

- success rate: 0.8600
- median(# queries): 4000

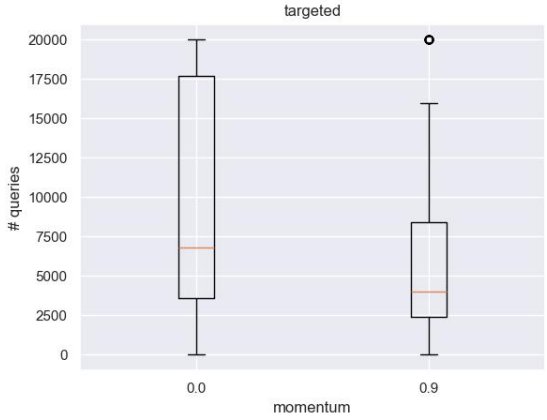

Figure 1: Black box # queries for targeted attack

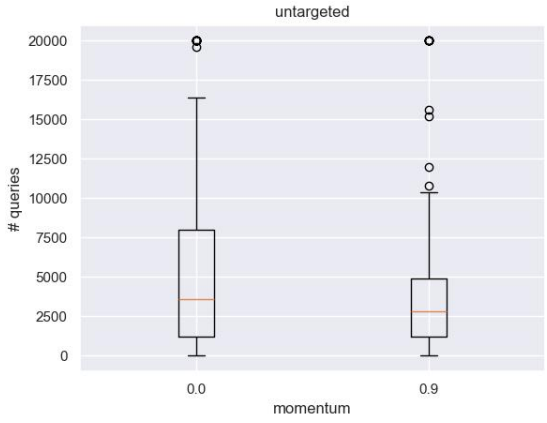

Figure 2: Black box # queries for untargeted attack

2 Transferability-based black-box attack

1. Here you will evaluate the transferability of untargeted and targeted PGD attacks between three pre-trained models (simple-cnn-0,1,2). Using the white-box PGD attack

you implemented in the previous question, run main_b.py to evaluate the transferability between the models. How well do targeted and untargeted attacks transfer?

Ans: Test accuracies:

- Model 0: 87.50%
- Model 1: 82.50%
- Model 2: 79.00%

Transferability matrices:

- Untargeted: $\begin{bmatrix} 0.98 & 0.56 & 0.53 \\ 0.70 & 0.965 & 0.585 \\ 0.60 & 0.55 & 0.955 \end{bmatrix}$
- Targeted: $\begin{bmatrix} 0.955 & 0.315 & 0.265 \\ 0.430 & 0.890 & 0.310 \\ 0.365 & 0.260 & 0.860 \end{bmatrix}$

It appears that untargeted attacks transfer more effectively than targeted ones across these model pairs.

Printed result from the python file:

Test accuracy of model 0: 0.8750

Test accuracy of model 1: 0.8250

Test accuracy of model 2: 0.7900

Untargeted attacks' transferability:

[[0.98 0.56 0.53]

[0.7 0.965 0.585]

[0.6 0.55 0.955]]

Targeted attacks' transferability:

[[0.955 0.315 0.265]

[0.43 0.89 0.31]

[0.365 0.26 0.86]]

2. To improve the success rate of transferred attacks, you will attempt to transfer adversarial examples produced against an ensemble of models, as proposed by Liu et

al. [3]. Fill out the missing code under attacks.PGDEnsembleAttack to implement PGD against an ensemble of models. Then, produce adversarial examples against models simple-cnn-1,2 and evaluate their success rate when transferred to simple-cnn-0. Did the transferability of untargeted and targeted attacks improve? If so, by how much and why?

Ans: When generating adversarial examples against the ensemble of models 1 & 2 and evaluating on model 0:

- Untargeted transfer success: 75.50% (up from 53–56%)
- Targeted transfer success: 52.50% (up from 26–31%)

The improvement arises because the ensemble gradient approximates a more general perturbation direction that affects multiple models simultaneously, enhancing cross-model robustness.

Printed result from the python file:

Ensemble attacks' transferability from models 1+2 to model 0:

- untargeted attack: 0.7550

- targeted attack: 0.5250

3 Bit-flip attacks

1. What is the maximum RAD?

Ans: The maximum Relative Accuracy Drop (RAD) observed is 73.33%.

2. What fraction of bits lead to ¿15

Ans: Approximately 2.84% of bit flips individually cause a RAD above 15%.

3. Examine the box-plot summarizing the RAD distribution for each of (the indices of) the 32 bits one can flip in a weight. Which bit has the highest median RAD? Why?

Ans: From the box-plot, the bit at index 1 has the highest median RAD. This is because it corresponds to the most significant quantization bit in the IEEE-754 representation (the highest-magnitude exponent bit), so flipping it induces the largest relative change in weight value.

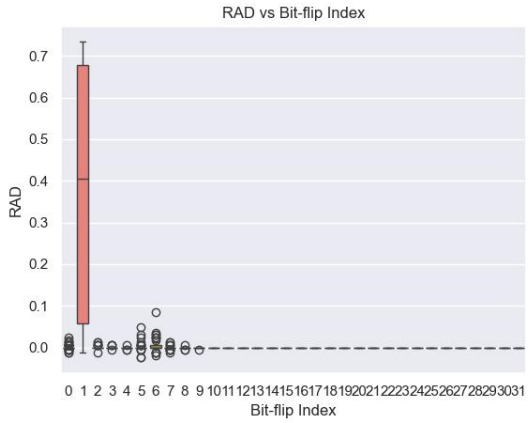

Figure 3: RAD distribution for each of (the indices of) the 32 bits one can flip in a weight