Writeup hw1

Submitted by: Yahav Lazar

ID – 213762180

<u>Question 1</u>

1. Based on the output the benign accuracy is 87.5%.
2. From the results(below), untargeted success rate: 0.9850, targeted success rate: 0.9400.
3. Black-box vs. White-box: The black-box rates are pretty good, but a bit lower than white-box (which were 98.5%/94.0%). That makes sense that estimating the gradient with NES obviously isn't as good as having the real thing.
   Did the Momentum help? Yeah, momentum definitely helped. It bumped up the success rates a bit for both attack types and it cut down the median queries needed by a lot (like ~1200 fewer for untargeted, ~2800 fewer for targeted, you can see this in the box plots too). Basically, momentum smooths out the inaccurate noisy gradient estimates from NES, helping it to stay in the same general directions and helping the attack find a working perturbation faster and better.

The raw main_a.py results:

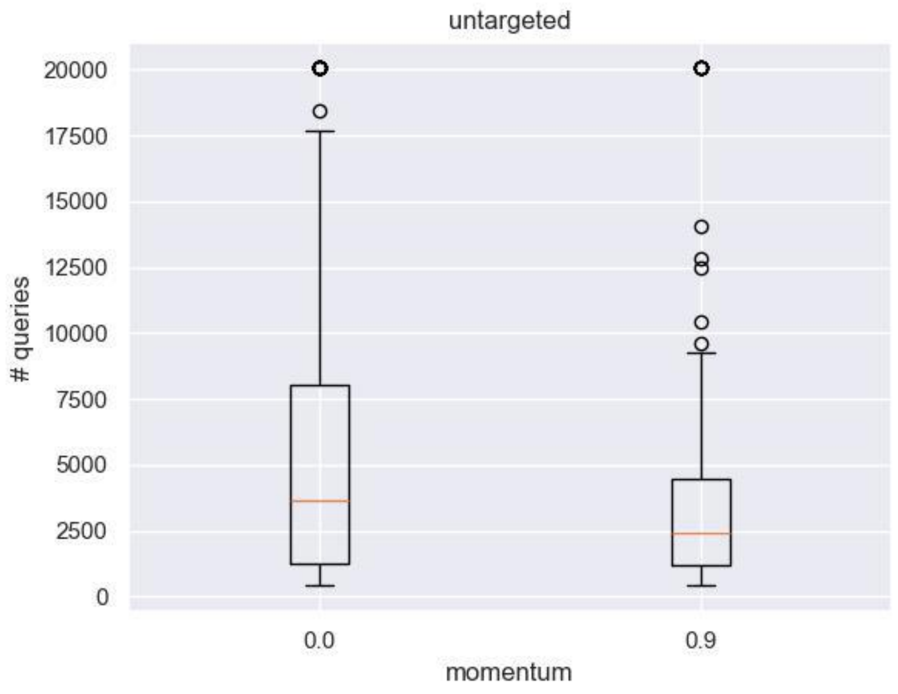
```
The test accuracy of the model is: 0.8750
White-box attack:
	- untargeted success rate: 0.9850
	- targeted success rate: 0.9400
Untargeted black-box attack (momentum=0.00):
	- success rate: 0.9250
	- median(# queries): 3630.0
Targeted black-box attack (momentum=0.00):
	- success rate: 0.7750
	- median(# queries): 6850.0
Untargeted black-box attack (momentum=0.90):
```

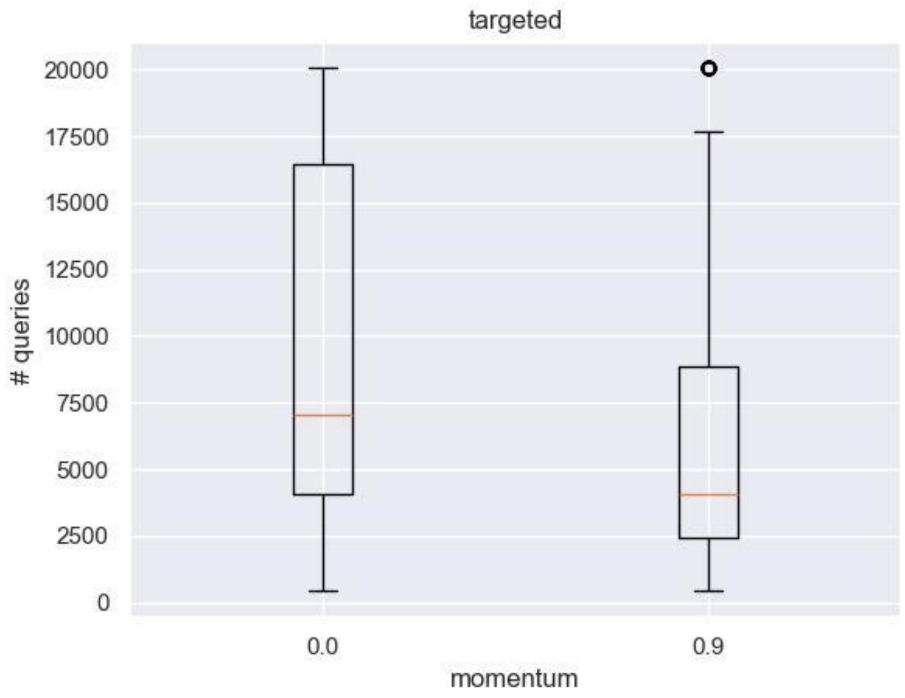
- success rate: 0.9650
- median(# queries): 2450.0

Targeted black-box attack (momentum=0.90):

- success rate: 0.8450
- median(# queries): 4050.0

The graphs:





Question 2

On the single model transfer, untargeted attacks transferred okay (50-70% success), while targeted attacks transferred pretty badly (25-42%). That makes sense cause targeted attacks are usually more model specific.

Did the ensemble improvement helped? Yep, the ensemble attack worked better for transfer. Untargeted went up to 74.0% (from a max of 70% single) and targeted went up to 50.0% (from a max of 42% single) when attacking model 0. It happened because attacking multiple models at once (the ensemble) makes the attack focus on more general weaknesses (and not overfit), which are more likely to exist in the target model too. And if the weakness worked on more models its more likely to work on ours as well if they trained on the same dataset because it may have something to do with the dataset weaknesses.

The raw main_b.py results:

```
Test accuracy of model 0: 0.8750

Test accuracy of model 1: 0.8250

Test accuracy of model 2: 0.7900

Untargeted attacks' transferability:

[[0.985 0.555 0.535]

 [0.7   0.965 0.595]

 [0.595 0.55  0.955]]

Targeted attacks' transferability:

[[0.94  0.335 0.265]

 [0.42  0.865 0.295]

 [0.355 0.275 0.845]]

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

        - untargeted attack: 0.7400

        - targeted attack: 0.5000
```

Question 3

1. maximum RAD: 0.7152
2. Fraction >15% RAD: 1.9%. Most flips had little effect. I think its due to the model needing to identify only 4 classes.
3. The highest median RAD Bit is Bit index 1 (from the box plot).
   Flipping the sign bit (index 0) changes positive to negative (or vice-versa). This can cause a large RAD if the original weight had a large magnitude, but if the weight was near zero, the change might be small.
   On the other hand, Bit 1 is the biggest exponent bit in the float representation so flipping exponent bits drastically changes the number's scale, impacting accuracy more than flipping small number signs or changing it by a little.

The raw main_c.py results:

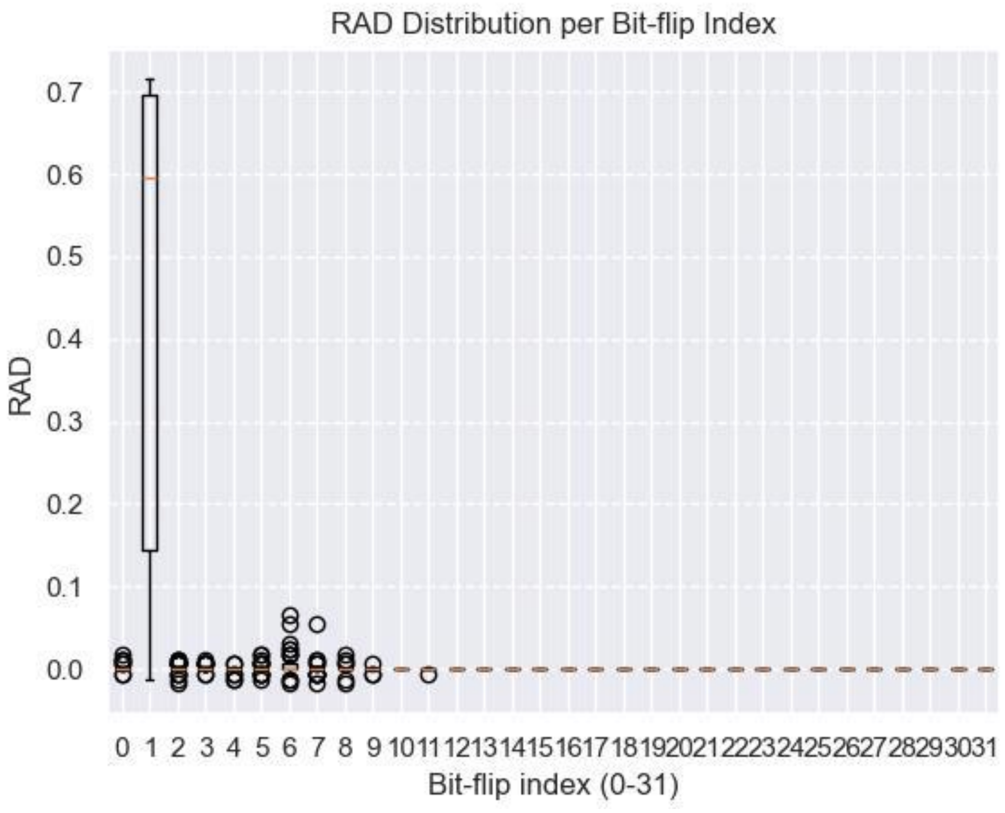
Model accuracy before flipping: 0.8250

Total # weights flipped: 2250

Max RAD: 0.7152

RAD>15%: 0.0191

The graph:


RAD Distribution per Bit-flip Index
RAD
Bit-flip index (0-31)