**Writeup hw2**

Submitted by: Yahav Lazar

ID – 213762180

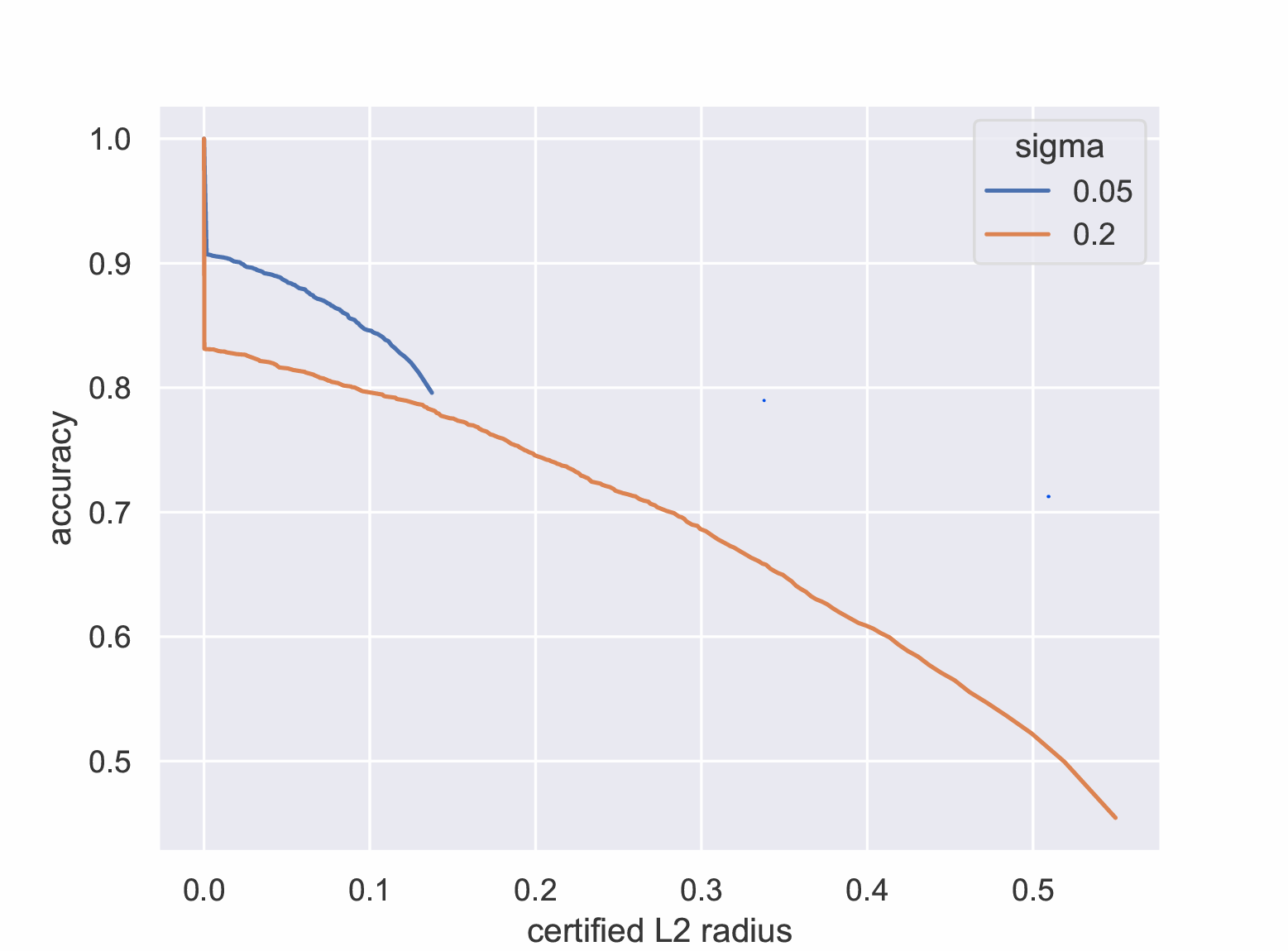
**Question 1**

Results:

|  |  |  |  |
| --- | --- | --- | --- |
| Training | Time(s) | Benign accuracy | PGD success rate |
| Standard | 3099.6511 | 0.9185 | 0.8995 |
| Adverserial training m=4 | 3518.0357 | 0.8965 | 0.2838 |
| Adverserial training m=7 | 3987.4953 | 0.8953 | 0.2723 |

It makes sense that the more we train adversarial training the worst the benign accuracy get and the more robust we are. Although there isn’t much of a difference between the m=4 and the m=7 in performance.

**Question 2**



Looking at the plot, a bigger σ gives you a larger certified radius, but the accuracy on clean inputs gets worse. The orange line for σ=0.2 goes much farther to the right, but it starts at a lower accuracy than the blue line for σ=0.05. This is because a larger σ means adding more noise during the process. More noise makes the model more robust, but it also hurts Benign accuracy.

**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.

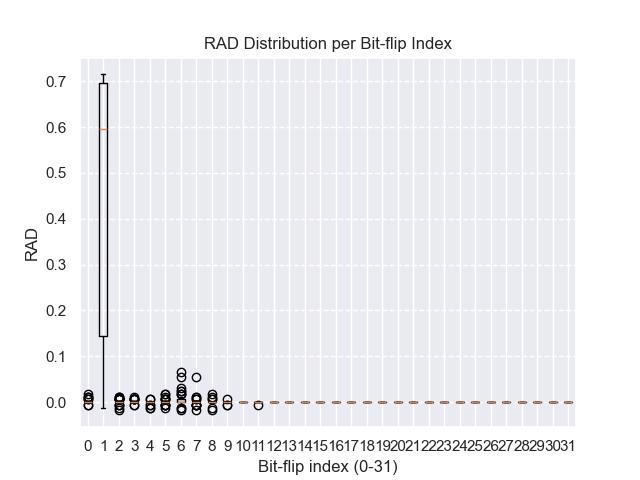
**The raw main\_c.py results:**

Model accuracy before flipping: 0.8250

Total # weights flipped: 2250

Max RAD: 0.7152

RAD>15%: 0.0191

**The graph:**