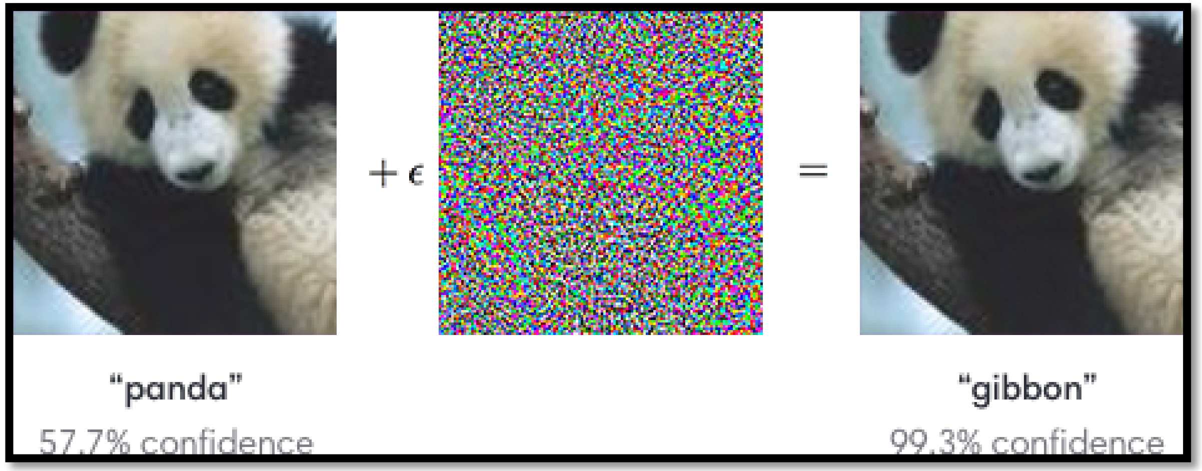
Detecting Adversarial Inputs at Runtime

Aymeric Fromherz
with Klas Leino, Matt Fredrikson, Bryan Parno, Corina Pasareanu

Machine Learning is vulnerable to attacks

- Wide variety of applications (Facial recognition, Autonomous cars, Malware detection, …)



- Vulnerable to "small perturbations"

Goal: Detect Adversarial Examples at Runtime

- **Local robustness:** For an input *x*, any "small perturbation" of *x* is classified as *x.*
- **Working hypothesis:** An adversarial example is not locally robust

Goal:

Decide Local Robustness at Runtime (i.e. quickly)

Previous Approaches: GeoCert, MIP, Reluplex

- Based on constraint-solving: Encode the network in its entirety, and solve queries exactly
- Already takes a few seconds on small MNIST networks
- Highly **precise**, but **slow** and **not scalable**

Our Approach: Fast Geometric Projections

- **Core idea:** Compute a lower bound on the distance to the closest adversarial point using **geometric projections**
- **Our tool:**
  - Certifies that a point is locally robust OR
  - Finds an adversarial example OR
  - Returns unknown
- Trade-off between precision and analysis speed

Experimental Results (with ε=0.25)

- Networks with Adversarial Training (median verification time)
  - MNIST with 3 layers of 20 neurons each: **0.02s**
  - MNIST with 9 layers of 20 neurons each: **0.67s**
  - MNIST with 3 layers of 40 neurons each: **2.13s**
- 100x to 10000x faster than best competitor (GeoCert)
- 2% to 7% unknown results
- Networks trained for verifiability (median verification time)
  - FMNIST with 20 layers of 100 neurons each: **0.08s**

Conclusion

- Faster local robustness certification, with low rate of unknowns
- Networks can also be trained for verifiability to increase scalability

Open questions:

- How can we improve further on scalability? Better training? Better verification heuristics?
- What other interesting properties of networks can we verify?

fromherz@cmu.edu