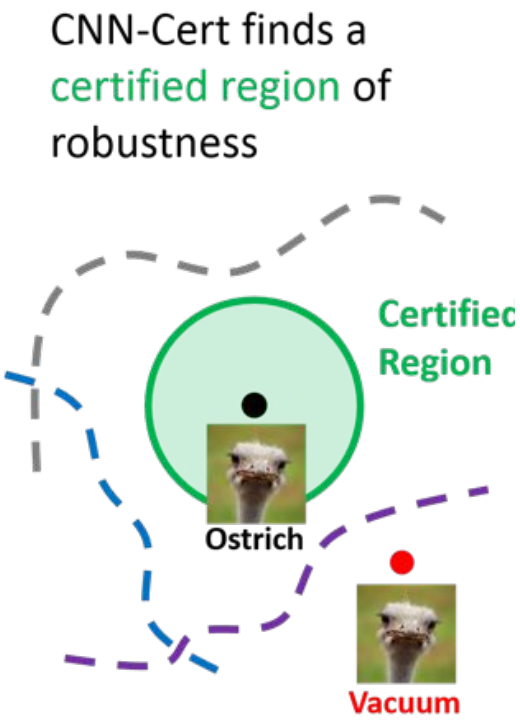
Generating Verifiably More Robustness CNN's

Carl Hildebrandt
Tyler Williams
Advait Kulkarni

Background

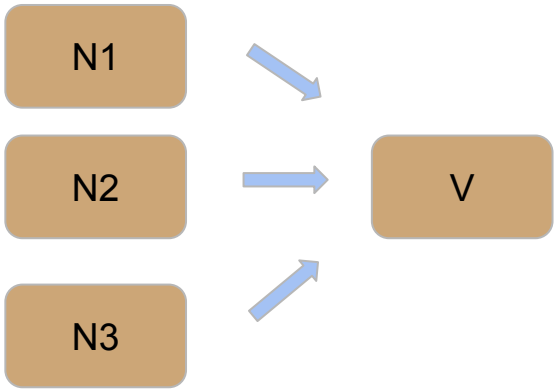
- We will be using the work in:
  - CNN-Cert: An Efficient Framework for Certifying Robustness of Convolutional Neural Networks

- This work outputs an Epsilon value which is the guaranteed region in which the test images can change without changing the networks output.



Design / Implementation / Simulation

- We will start with a single network **A**.
- We will run CNN-Cert on **A** to calculate its robustness.
- We will create three different networks which we will connect to a voter in a TMR setup.
- We will use this network as a teacher for a single network **B** which has the same architecture as **A**.
- We will run CCN-Cert on **B** to calculate its robustness.

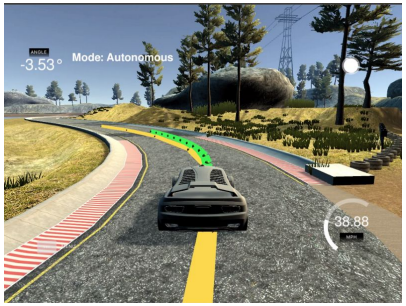


Teaches:

The hope is we can learn the robustness which occurs due to TMR.

Measuring Data

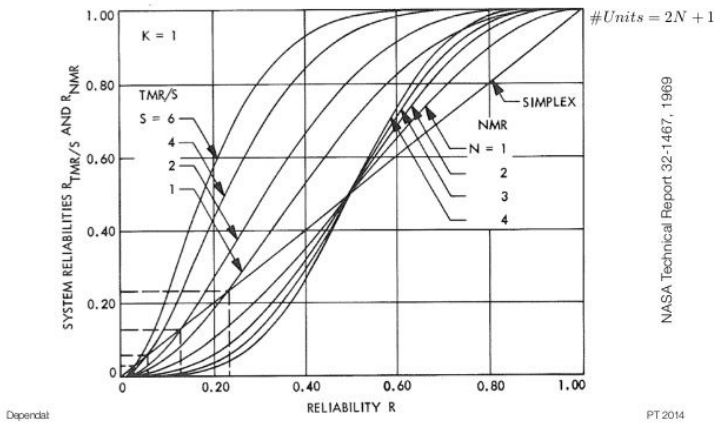
- CCN-Cert gives an epsilon value which is the L-Norm distance.
- We will also run real tests using foolbox described in:
  - Foolbox: A Python toolbox to benchmark the robustness of machine learning models
- This will allow us to generate adversarial images using different adversarial techniques.
- We will also look into running real networks such as self driving networks from Udacity.



Concepts Applied

- We will be applying techniques from reliability. The main technique we will be using is TMR.
- It could be interesting to apply other techniques:
  - Dual System
  - N-Version Programming
  - Simplex
- These will allow us to demonstrate the benefits of having more or fewer networks

Comparison TMR vs. TMR/S vs. NMR



Dependat

PT 2014

Timeline + Task Allocation

Key

- Carl
- Tyler
- Advait