Summary of Session 2: Self-Driving Cars and Safety

Domenico Cotroneo

Two presentations on the two extremes of the problem

From the hardware

1. **Paolo Rech,** Can We rely on Self-Driving Cars? Evaluation and Mitigation of Neutron-Induced Errors in Convolutional Neural Networks for Autonomous Vehicles

To Development Process (Safety engineering)

2. **Daniel Schneider**, Safety-critical systems with Machine Learning components: Challenges and solutions

Paolo Rech,
Can We rely on Self-Driving Cars? Evaluation and Mitigation of Neutron-Induced Errors in Convolutional Neural Networks for Autonomous Vehicles

Focused on

- The analysis of Neutrons-induced effects in computing devices
  - Neutron-induced faults may induce Application Crash or Device Reboot
- Cross layer faults propogation in CNNs
  - Observed errors in the autonomous vehicles context
    - False postives (e.g., unnecessary stops)
    - Classification errors (e.g., wrong object detection)
  - Faults propogation is a non-trivial task for the current (complex) devices
    - FI, as it is, seems to be not the right solution (simplistic fault model)
- Proposed solutions
  - Algorithm-based Fault tolerance (corrects 87% of critical SDC errors)
  - Max/smart pooling
  - Mixed-precision hardening (detection improvement)

Paolo's conclusions

Can we rely on Self-Driving Cars?

- Not all faults reach the software level
  - Not all errors are critical for CNNs
  - How many errors impact the vehicle behavior?
- The fault model is not naive in modern architectures
  - Realistic fault model is necessary to design effective hardening
- The corrupted value(s) depend(s) on several variables

….maybe FI shpould be rethinked/extended to be applied in this domain

Daniel Schneider,
Safety-critical systems with Machine Learning components: Challenges and solutions

- Challenge on the use of ML component in safety CPS
  - In the context of Auotonomous vehicles the use of AI is critical and it is characterized by dynamic learning
- ML engineering : Integrated Safety and ML engineering e.g.,
  - Only use ML components when there is no acceptable conventional solutions
  - Need of methods and of techniques for analyzing and hardening
  - Robustness assurance of learned model
  - Validation as central element of assurance
- Dynamic Risk Management (DRM) Vision
  - Model based safety engineering at **Design time** vs Dynamic risk management at **Runtime**
  - DRM runtime Architecture

Daniel' vision

- Uncertainty caused by
  - limitations of the learned model
  - Limitations during the model application
  - Mismatch between target/test context and application context
- SafeML is valuable to evaluate 'how far' from our trained context we are currently operating

- The conclusions:
  - There is no single silver bullet for assuring safety of systems with ML-components
  - There is no commonly accepted state of the practice or even a sound understanding with respect to suitable engineering methods, techniques and tools.
  - *General solution approaches, recommendations DRm, dealing with uncertainty*

….how to control the dynamicity of risks