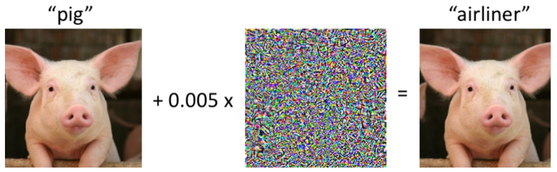
Speeding Up Neural Network Verification via Automated Algorithm Configuration

Self-monitoring AI systems

Matthias König, Holger H. Hoos, Jan N. van Rijn ▪ Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands

Background

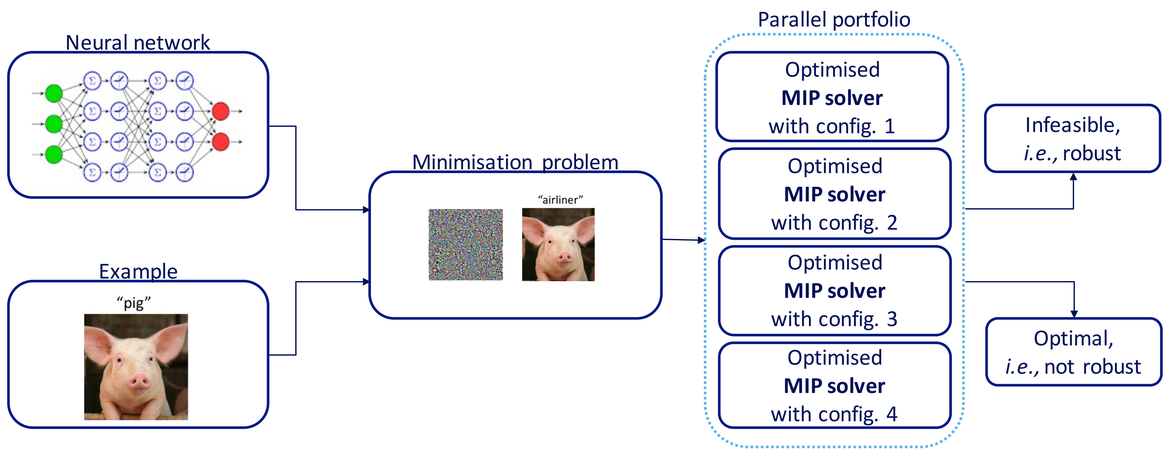
- Neural networks are vulnerable to adversarial examples



- Several neural network verification methods are based on mixed integer linear programming (MIP)
- Problem: High computational costs and many timeouts

Method

- Parallel portfolio of optimised solver configurations
- 1 CPU core per configuration



Conclusions

- Improved performance of state-of-the-art MIP-based verification engine MIPVerify:
  - 4.7-fold reduction in CPU time
  - 1.4 times fewer timeouts
  - 1.3-fold improvement on upper bound
- Future work considers further MIP-based verification engines, classifiers and datasets

Baselines & Results

- Task: Verify robust classifier over full MNIST dataset (n=10 000)
- Baselines: MIPVerify verification engine with Gurobi solver at default using *(i)* all available CPU cores, *(ii)* 4 CPU cores, *(iii)* 1 CPU core with same overall CPU time budget
- Avg. running time on subset of *solvable* instances (instances solved by any approach; n=8 646)

	Portfolio (4 Cores)	Default** (32 Cores)	Default (4 Cores)	Default (1 Core)
Timeouts	**14.96%**	21.29%	17.74%	17.66%
Adversarial error Upper bound	**23.86%**	30.67%	27.49%	27.58%
Lower bound	14.43%	14.37%	14.40%	14.36%
Avg. running time [CPU s]*	**8 478**	39 772	22 065	20 117

* Timeouts are penalised as t x 10.
** As employed in MIPVerify