Synchronous Artificial Neural Networks for Safety Critical Systems

Keyan Monadjem

August 17, 2018

Overview

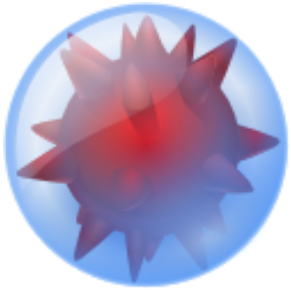
What is Artificial Intelligence?

The aim for machines to intelligently decide the best course of action to meet their respective goals.

Machine-based...

- Acquisition and Manipulation of Knowledge
- Generation and Achievement of Goals

Are they useful?



Security

Finance

Analytics

Banking

Automotive

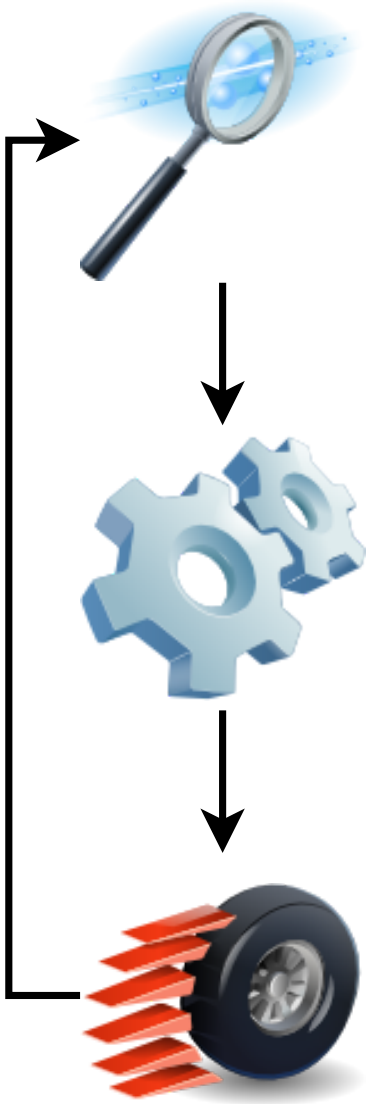
Image/Pattern Recognition

There are many kinds of AI

(Some) Types

- Symbolic AI
- Statistical Learning
- Sub-symbolic
  - Evolutionary Computation
  - Probabilistic Modelling
  - **Artificial Neural Networks (ANNs)**

AI is *reactive*...

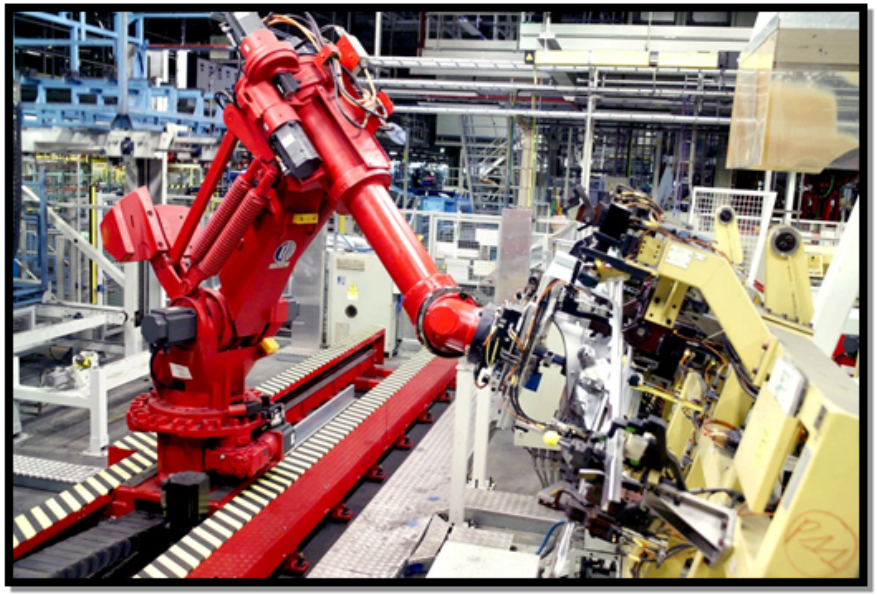


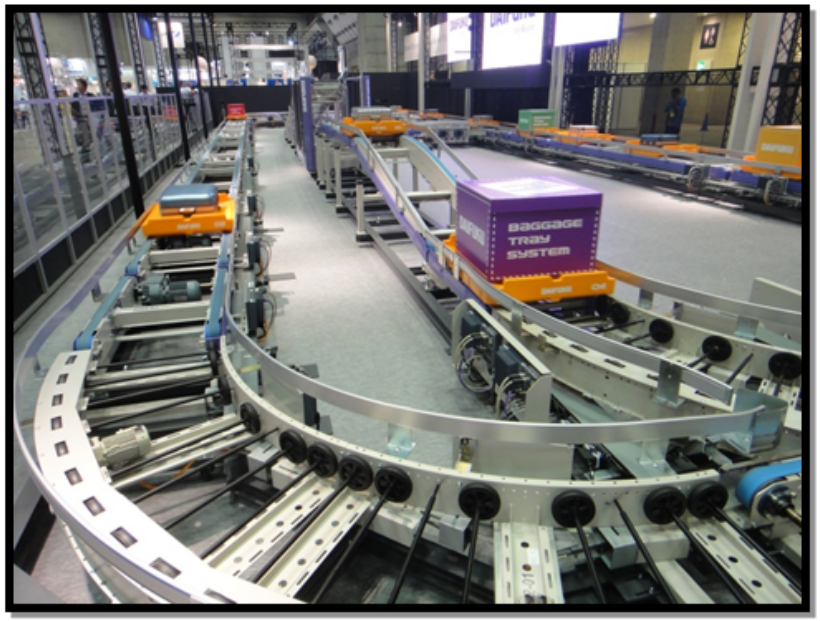
1. Perceive Environment

2. Analyse Findings

3. Act in interest of goal

Reactive Safety-Critical Systems





Overview

A brief overview of ML

Software that learns data relationships without the software knowing the relationships beforehand.

- Decision trees
- **ANNs - deep learning**
- Reinforcement learning

A brief overview of ANNs

- Originally designed to model the brain.
- One neural network is a group of interconnected nodes, called artificial neurons, which pass signals among themselves.

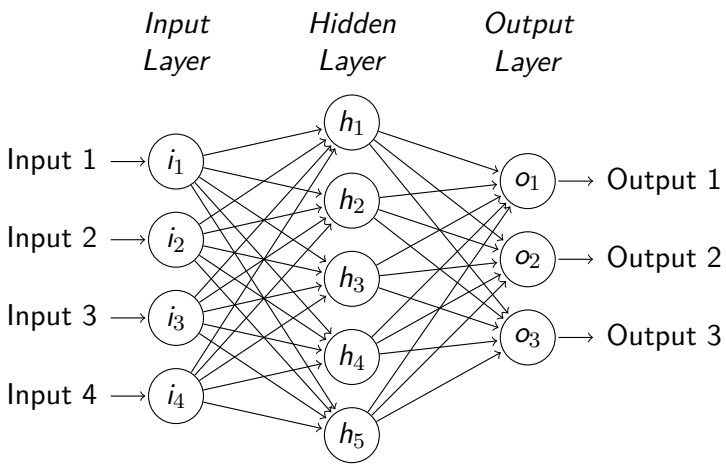


Figure: Example of an ANN.

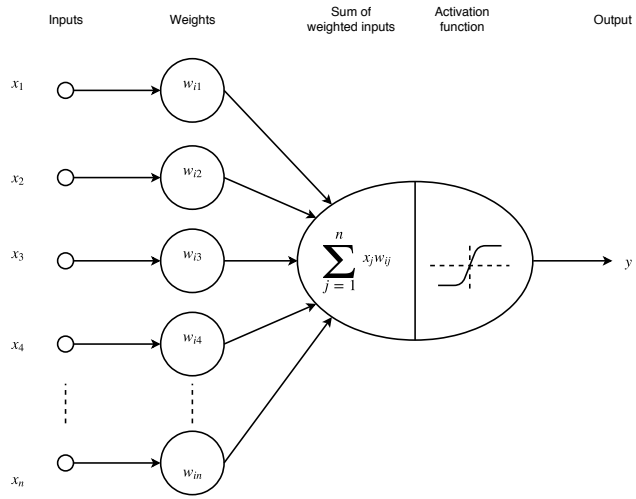


Figure: Example of an artificial neuron.

Verified ANNs for safety critical systems

- Software/systems based on ML can be very complex.
- Not suitable for safety critical systems without validation/verification.
- Existing techniques to verify/validate ANNs for safety critical environments (proactive) - not always ideal.
- Few solutions for *reactive* safety critical ANNs.
- Functional Analysis and formal methods (e.g. **WCRT**).
- **Runtime Enforcement**.

Overview

Current AI in safety-critical applications

Validation and Verification

- Unit testing
- Strict guidelines
- Verification algorithms[1][2][3]

[1] Xiaowei Huang et al. "Safety Verification of Deep Neural Networks". In: *Computer Aided Verification*. Ed. by Rupak Majumdar and Viktor Kunčak. Cham: Springer International Publishing, 2017, pp. 3–29. ISBN: 978-3-319-63387-9.

[2] Guy Katz et al. "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks". In: *Computer Aided Verification*. Ed. by Rupak Majumdar and Viktor Kunčak. Cham: Springer International Publishing, 2017, pp. 97–117. ISBN: 978-3-319-63387-9.

[3] T. Gehr et al. "AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation". In: *2018 IEEE Symposium on Security and Privacy (SP)*. May 2018, pp. 3–18. DOI: 10.1109/SP.2018.00058.

Safety Critical Artificial Neural Networks (SCANNs)[4]

- Fuzzy Self-Organising Maps (FSOMs)
- Rule extraction and insertion
- Safety cases

[4] Zeshan Kurd. "Artificial Neural Networks in Safety-critical Applications". PhD dissertation. University of York, 2002.

Why Are Current Implementations *Bad*?

- Difficult to formally quantify an ANN.
- Verification of ANNs hot topic and highly researched.
- Does not address faulty run-time outputs.

Overview

Uber autonomous vehicle fatal accident

What happened?

[5]

- Uber autonomous vehicle.
- Vehicle collided with jaywalking pedestrian.
- **Fatal**.

[5] A. C. Madrigal. "Uber's Self-Driving Car Didn't Malfunction, It Was Just Bad". In: *The Atlantic* (May 2018).

What went wrong?

- Pedestrian was sufficiently inebriated.
- ANN misclassified the pedestrian **multiple times**.
- Software (AI) trusted over LiDAR readings.
- Vehicle did not brake and driver was not paying attention.
- Failure on many fronts, including the software of a *safety critical* system.

Problem Statement

How can artificial neural networks utilise synchronous languages such that the safety and performance of safety-critical systems is improved?

Overview

Introduction to Synchronous Semantics[6]

- No chance of deadlocks occurring during runtime.
- Loops are bounded.
- Causality is maintained.
- System is deterministic.
- Runtime enforcement and observers.
- Easier to formally analyse Worst Case Execution Time (WCET) and Worst Case Reaction Time (WCRT).

[6] Albert Benveniste et al. "The synchronous languages 12 years later". In: *Proceedings of the IEEE* 91.1 (2003), pp. 64–83.

WCET

- Blackbox ANN that "runs fast enough" - not good enough.
- WCET of ANNs using formal methods largely unexplored.
- WCET analysis is usually measurement-based .

Synchronous Artificial Neural Networks (SANNs)

- Created upon the basis of synchronous semantics using Esterel.
- Receives all the bonuses of a synchronous language.
- Various ANN compositions, both inter-network (meta neural networks) and intra-network (network scheduling).

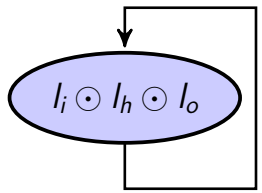


Figure: Black box execution of an SANN.

Black-box scheduling

- Single function call to run the entire neural network in one synchronous tick.
- Inputs are provided, outputs are returned.

Disadvantages

- Function calls from synchronous programs are hardly new.
- Iterates through all layers of the Neural Network in one tick - slow.
- High WCRT should the system fail.

Layer by Layer scheduling

- Each synchronous tick runs one layer of the neural network
- Between layer results are stored to be used in the next tick
- Faster WCRT than black box - system can respond between layers

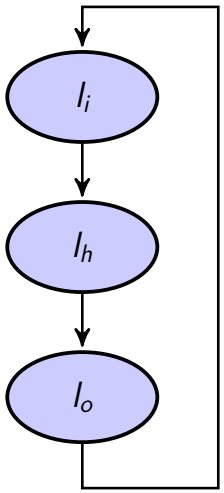


Figure: Layer by layer execution of an SANN.

Neuron by Neuron scheduling

- Each artificial neuron created in Esterel.
- Each synchronous tick runs one layer of the neural network.
- Each layer connected by signals to previous and/or following layers.
- Faster WCRT than layer by layer scheduling - less overhead.

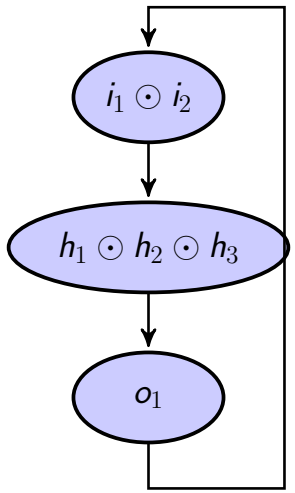


Figure: Neuron by neuron execution

Meta Neural Networks

- A composition of SANN black boxes or otherwise.
- Intricate combinations feasible due to synchronous semantics.

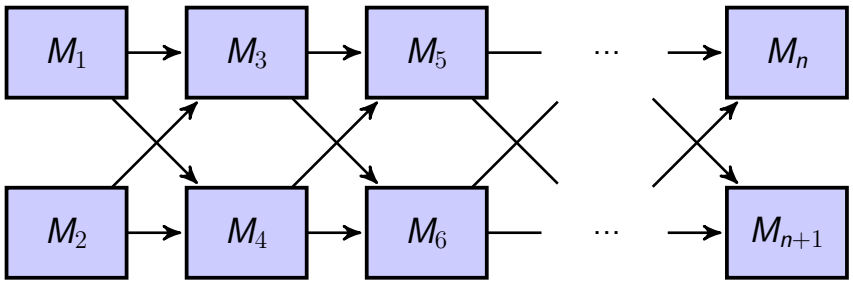


Figure: Example meta neural network with parallel SANNs.

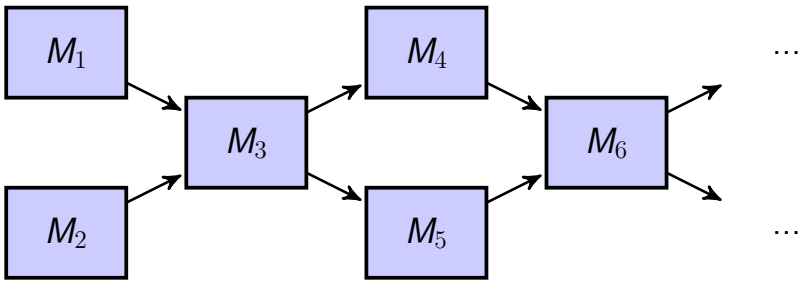


Figure: Example meta neural network with some parallel SANNs.

Runtime Enforcement[7]

- Monitors and corrects a system's I/O.
- Cannot delay events or block execution.
- E.g. enforce that the vehicle will brake, instead of accelerating, when a collision could occur.

Observers

- Statically specify properties of a program.
- Verification for synchronous languages.

[7] Srinivas Pinisetty et al. "Runtime Enforcement of Cyber-Physical Systems". In: *ACM Trans. Embed. Comput. Syst.* 16.5s (Sept. 2017), 178:1–178:25. ISSN: 1539-9087. DOI: 10.1145/3126500. URL: http://doi.acm.org.ezproxy.auckland.ac.nz/10.1145/3126500.

Safe Neural Networks (SNNs)

- Synchronous Artificial Neural Network (SANN) encapsulated by Runtime Enforcement (RE).
- SANN I/O monitored and enforced to be *safe*.
- SNN is synchronous and

Safe Neural Networks (SNNs) - Definition

Definition 1

An SNN is formalised as a tuple $S = \langle I, O, \varphi_I, \varphi_O, L, \gamma, \alpha \rangle$, where:

- I is a finite collection of input variables with its domain being $\mathbf{I} = \mathbb{R}^n$.
- O is a finite collection of output variables with its domain being $\mathbf{O} = \mathbb{R}^m$.
- φ_I is an input safety policy specifying the safe behaviour of inputs I.
- φ_O is an output safety policy specifying the safe behaviour of input-outputs $I \times O$.
- L denotes a set of ANN layers $\{l_1, l_2, ..., l_k\}$.
- $\gamma : I_l \rightarrow O_l$ is the non-linear activation function that provides the behaviour of a given layer, i.e. when provided inputs I_l, the layer produces outputs O_l.
- $\alpha : l_k \rightarrow l_{k+1}$ is the layer-to-layer mapping function that maps the outputs of a given layer to the inputs of the following layer.

Safe Neural Networks (SNNs) - AV as a SNN

- $I = \langle S, P, O_1, O_{1_S}, O_{1_D}..., O_5, O_{5_S}, O_{5_D} \rangle$, i.e. the 17 different fixed-point integer inputs to the controller ANN.
- $O = \langle A, B_S, B_H \rangle$ are the three different fixed-point integer outputs which represent the different actions (and the confidence in the action) that can be taken by the AV at any given time.
- $\varphi_I = \varphi_{av_I} = \varphi_{cnn_I} \wedge \varphi_{ped_I} \wedge \varphi_{car_I} \wedge \varphi_{drive_I}$, i.e. the complete set of policies projected on inputs I. Once combined, they ensure the safety of the CNN ensemble and LiDAR outputs (i.e. the controller's inputs).
- $\varphi_O = \varphi_{av} = \varphi_{cnn} \wedge \varphi_{ped} \wedge \varphi_{car} \wedge \varphi_{drive}$, i.e. the complete set of policies ensuring the safety of the controller. Once combined, they prevent collisions with pedestrians, other vehicles and maintain that the car drives consistently and within the law.
- $L = \{l_i, l_h, l_o, l_{pp}\}$, are the four layers of the controller ANN, i.e. the input layer, hidden layer, and output layer of the MLP; and the post processing layer of the controller.

Overview

System Simulation: Autonomous Vehicle (AV)

- Enforcer lies between the controller and the plant.
- Plant consists of sensors and actuators.
- Sensors: LiDAR and cameras
- Actuators: Motors and brakes (drive and stop)

AV System Sensor Diagram

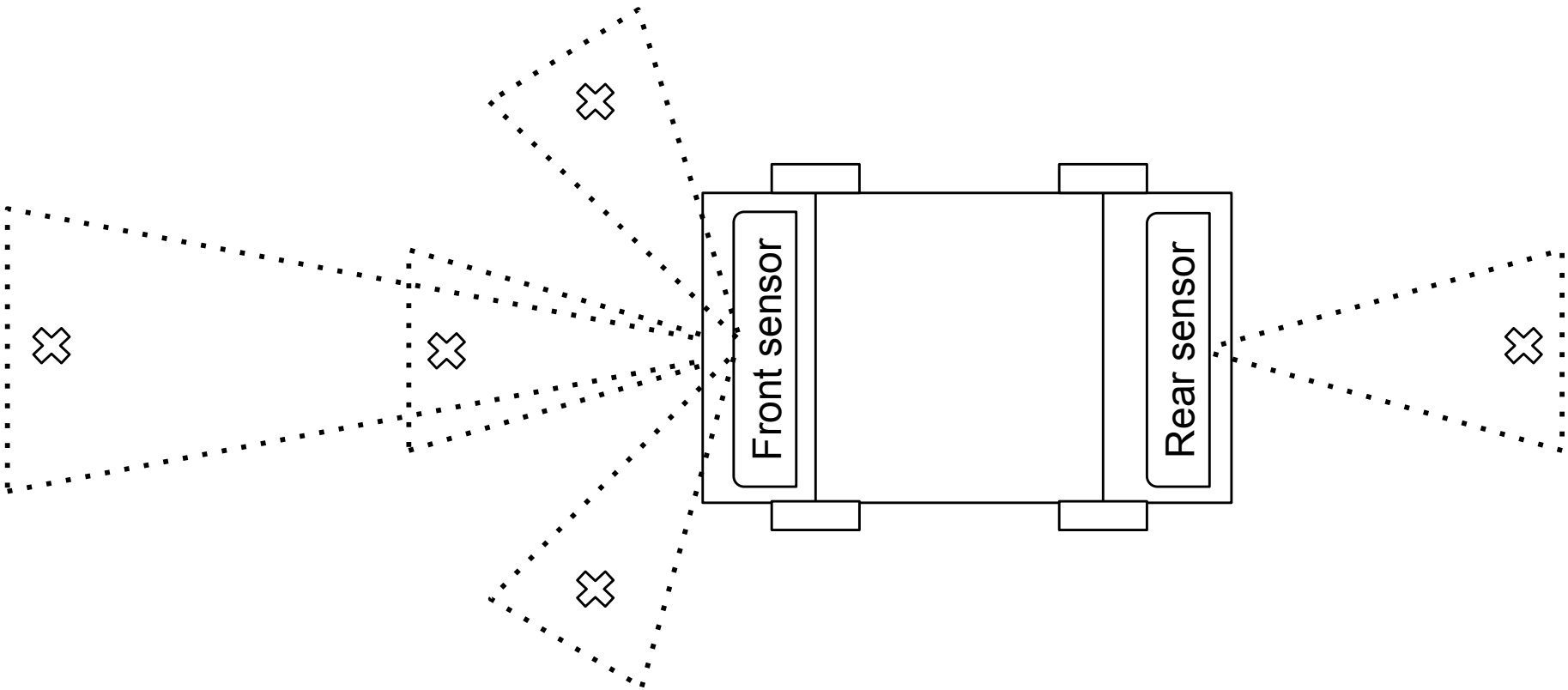


Figure: Sensor diagram for the AV safety system.

AV System Block Diagram

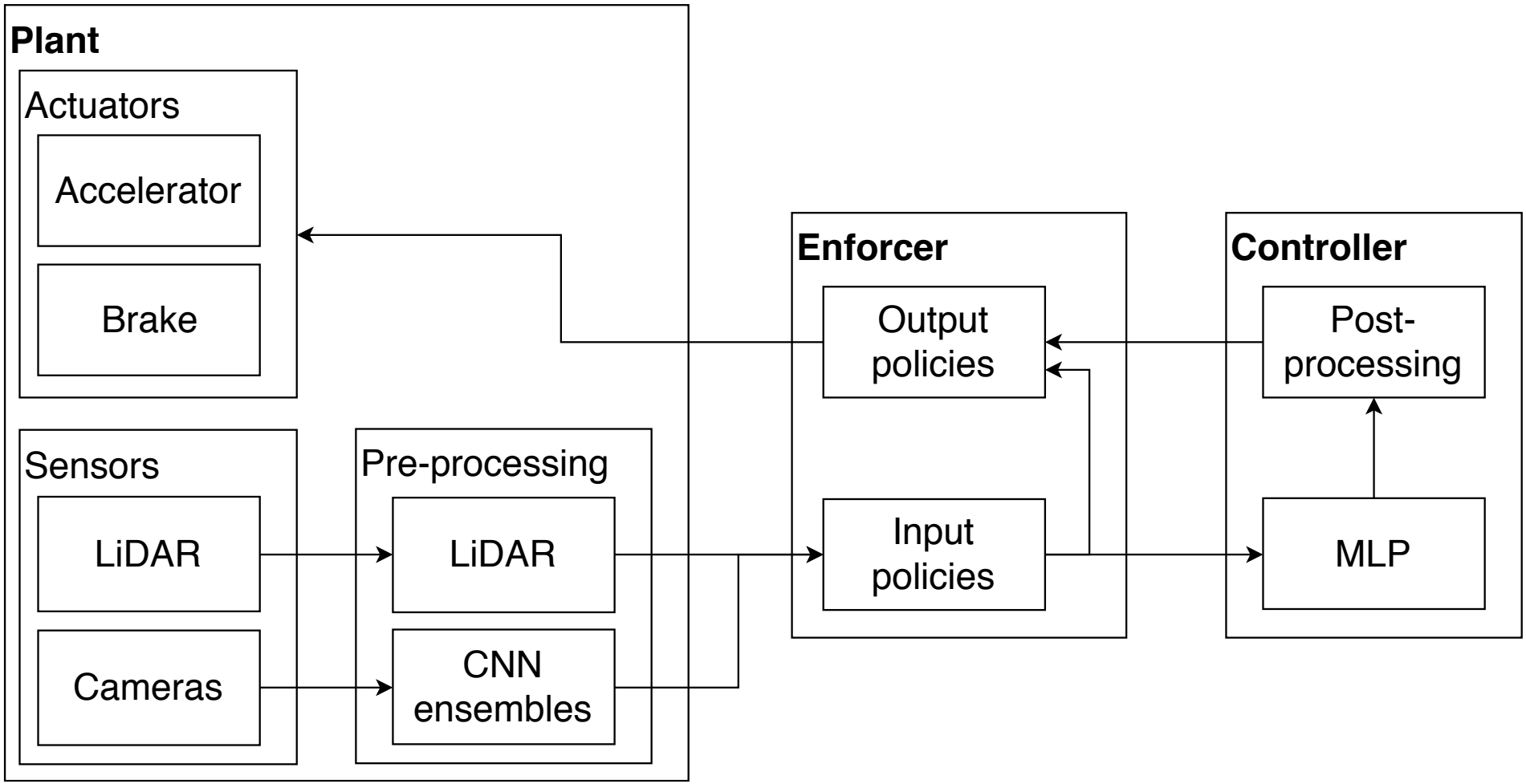


Figure: Block diagram for the AV safety system.

Runtime Enforcement

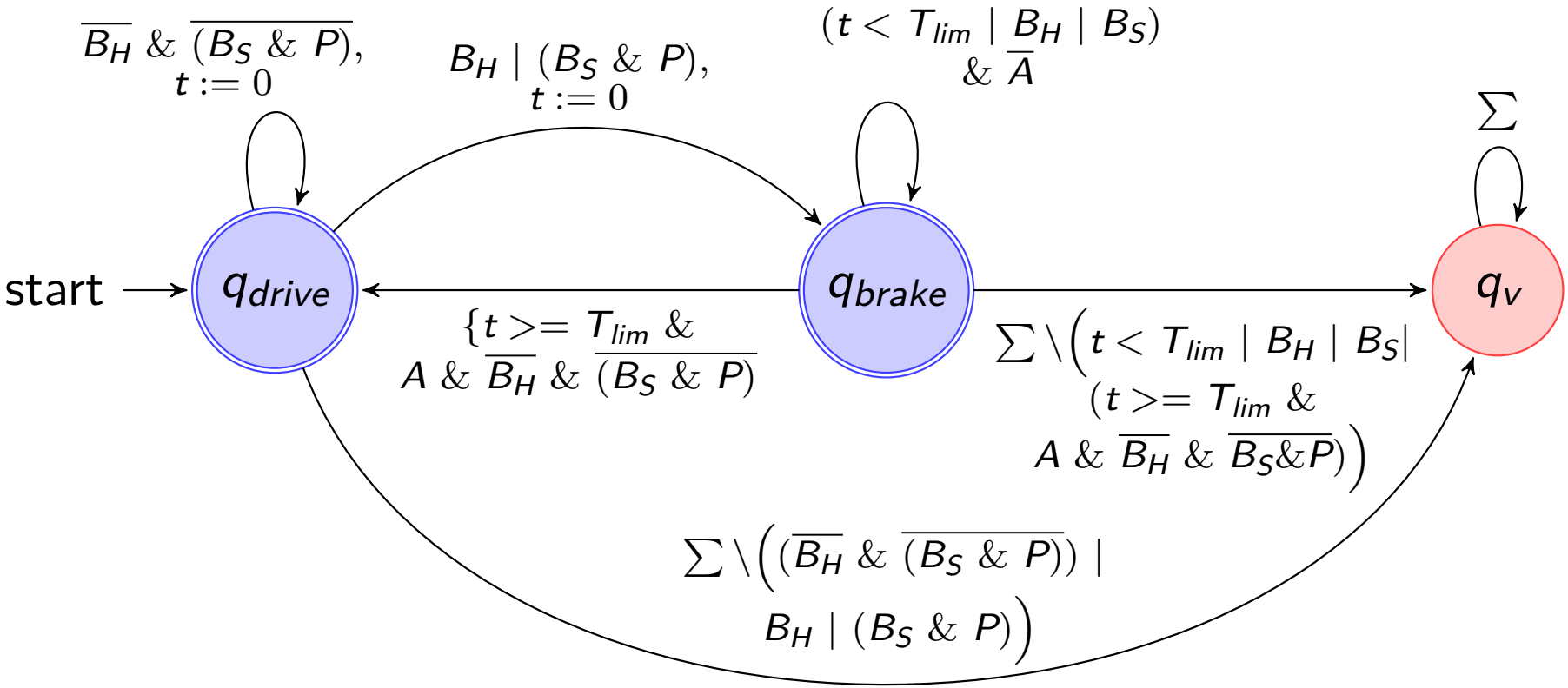


Figure: Basic RTE state machine for the AV safety system

Results

Table: Design and overhead of the policies used in the AV system

Policy	States	Transitions	Timed	Execution Time (us)	Overhead (%)
None				736	0
φ_{cnn}	1	15	No	764	3.8
φ_{drive}	1	4	No	740	0.54
φ_{car}	1	7	No	774	5.1
φ_{ped}	2	56	Yes	767	4.2
$\varphi_{cnn} \wedge \varphi_{drive} \wedge \varphi_{car} \wedge \varphi_{ped}$	2	99	Yes	803	9.1

Results (cont.)

Table: Results of the AV system with and without the enforced policies

Number of epochs trained	Percentage accidents	Average minutes to first accident per day	Average speed (km/h)	Percentage bad brakes
Not enforced				
0	100	3.21	98	19
1	100	3.3	93	19
10	100	3.8	81	57
100	100	5.35	59	24
1000	100	5.31	58	27
10000	97.5	25.3	30	70

Results (cont.)

Table: Results of the AV system with and without the enforced policies

Number of epochs trained	Percentage accidents	Average minutes to first accident per day	Average speed (km/h)	Percentage bad brakes
Enforced				
0	757	756	45	0.7
1	702	828	47	4.2
10	885	589	37	20
100	795	726	41	17
1000	805	710	40	18
10000	636	910	30	27

Results (cont.)

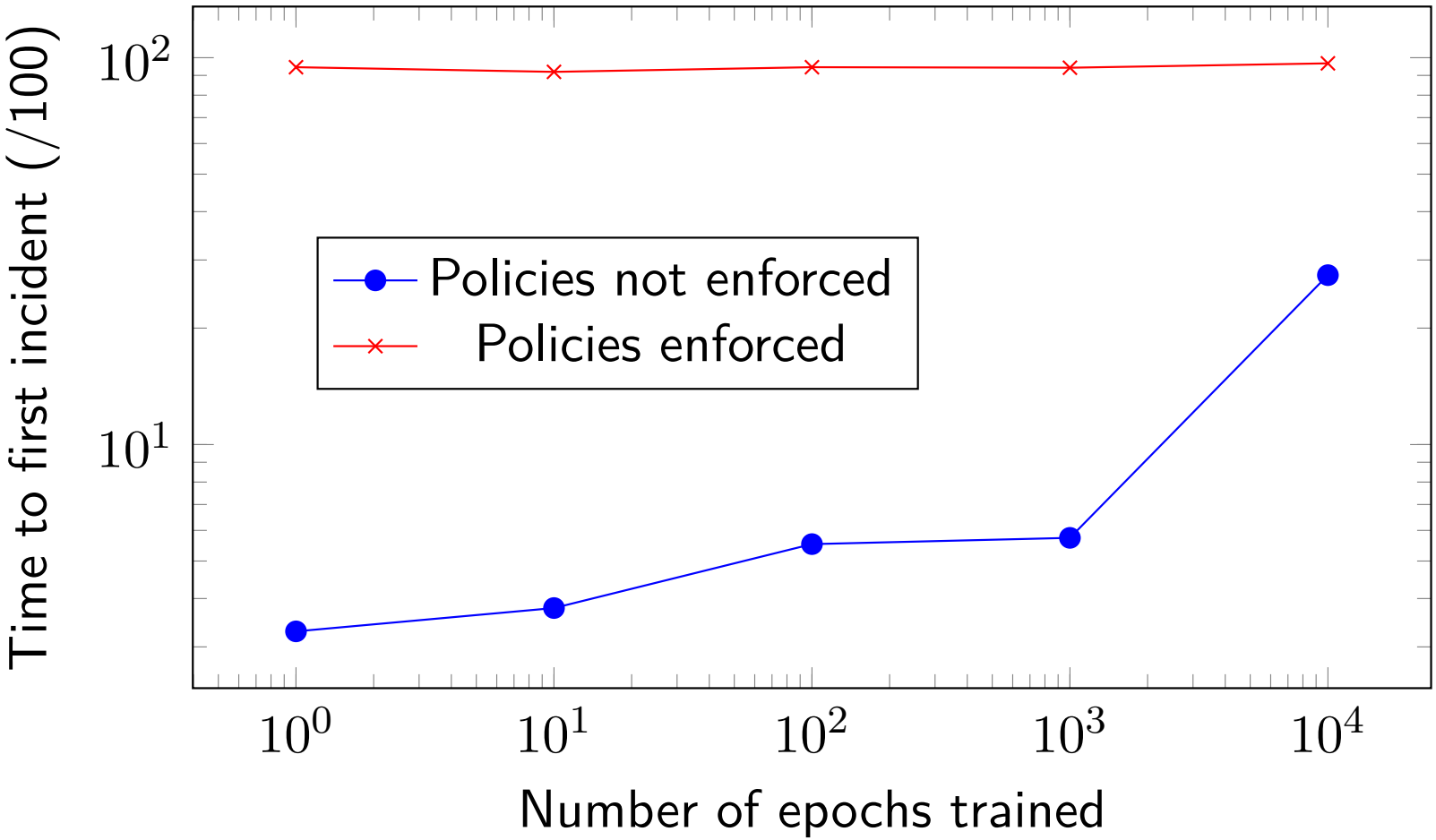


Figure: Line graph showing the performance of the enforced system compared to the un-enforced system

Overview

Autonomous Vehicle (AV) working example

AV system

- Lots of TA marking...
- Training system with an increased number of epochs.
- Step from 10,000 to 100,000.

Paper for IEEE journals

TO-DO

-

Overview

WCRT

- WCRT tests were carried out on various, synchronous systems with SANNs.
- ABRO example utilizing SANNs shown below.

Table: WCRT results for AI version of ABRO

Approach	WCRT (ms)	WCET (ms)
Black-box	2.8	2.8
Layer by layer	1.9	7.6
Neuron by neuron	1.8	7.2

Meta neural networks

- Using an ensemble of meta neural networks[8], it was shown that the combined accuracy of 3 SANNs was greater than the accuracy of any one SANN.
- Increase of up to 10% from individual SANN accuracy.
- Retain synchronous semantics.
- Potential for WCRT analysis.

[8] L. K. Hansen and P. Salamon. "Neural network ensembles". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.10 (Oct. 1990), pp. 993–1001. ISSN: 0162-8828. DOI: 10.1109/34.58871.

Runtime enforcement

- TBD.

Overview

Conclusion

-

Benveniste, Albert et al. “The synchronous languages 12 years later”. In: *Proceedings of the IEEE* 91.1 (2003), pp. 64–83.

Gehr, T. et al. “AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation”. In: *2018 IEEE Symposium on Security and Privacy (SP)*. May 2018, pp. 3–18. DOI: 10.1109/SP.2018.00058.

Hansen, L. K. and P. Salamon. “Neural network ensembles”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.10 (Oct. 1990), pp. 993–1001. ISSN: 0162-8828. DOI: 10.1109/34.58871.

Huang, Xiaowei et al. “Safety Verification of Deep Neural Networks”. In: *Computer Aided Verification*. Ed. by Rupak Majumdar and Viktor Kunčak. Cham: Springer International Publishing, 2017, pp. 3–29. ISBN: 978-3-319-63387-9.

Katz, Guy et al. “Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks”. In: *Computer Aided Verification*. Ed. by Rupak Majumdar and Viktor Kunčak. Cham: Springer International Publishing, 2017, pp. 97–117. ISBN: 978-3-319-63387-9.

Kurd, Zeshan. “Artificial Neural Networks in Safety-critical Applications”. PhD dissertation. University of York, 2002.