# Formal Methods in Machine Learning

## A 30000-feet view

## **(Why CS781?)**

Supratik Chakraborty
IIT Bombay

# Safety in AI/ML

- AI/ML based systems
  - Computational systems that try to mimic (and improve upon?) human reasoning
- Applications span entire spectrum of consequences
  - **Benign**
    - Auto completion in chat, game of chess, recommendation of restaurants, …
  - **Potentally serious, but recoverable**
    - Approval of bank loans, bail applications, ...
  - **Serious irrecoverable consequences**
    - Collision avoidance in unmanned drones, self-driving cars, malware detection, …

- **Can we trust decisions by AI/ML based systems in applications where cost of errors is extraordinarily large?**
  - **Human lives, breach of privacy, security gaps, loss of critical infrastructure …**

# Something Requires Attention …

10 Lessons From Uber's Fatal Self-Driving Car Crash

The fallout from the first autonomous car fatality continues to swirl a year later, as the once high-flying technology faces a "trough of disillusionment."

BY EDWARD NIEDERMEYER | UPDATED MAY 17, 2019 12:58 PM

NEWS

JAN 11

by NOOR AL-SIBAI

/ ADVANCED TRANSPORT

FAIL SELF DRIVING

SELF-DRIVING TESLA CAUSES EIGHT-VEHICLE CRASH, INJURES CHILD

THIS IS NOT GOOD. THIS IS NOT GOOD AT ALL.

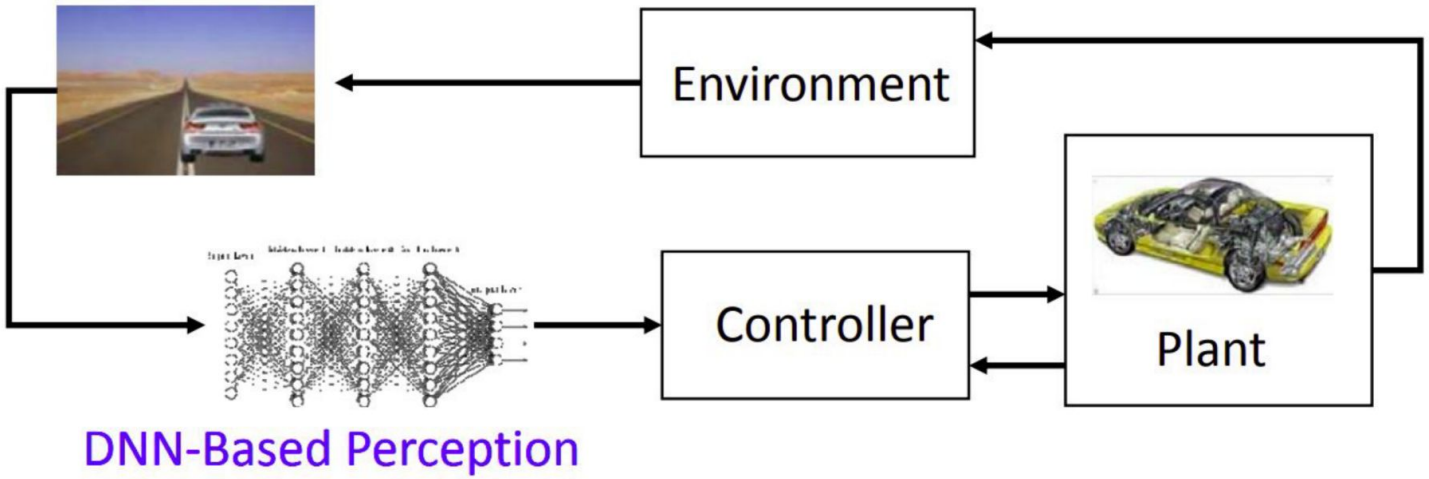
TC

**Drone crash near kids leads Swiss Post and Matternet to suspend autonomous deliveries**

Devin Coldewey @techcrunch / 5:47 AM GMT+5:30 • July 31, 2019 Comment

Image Credits: Matternet Inc. under a license.

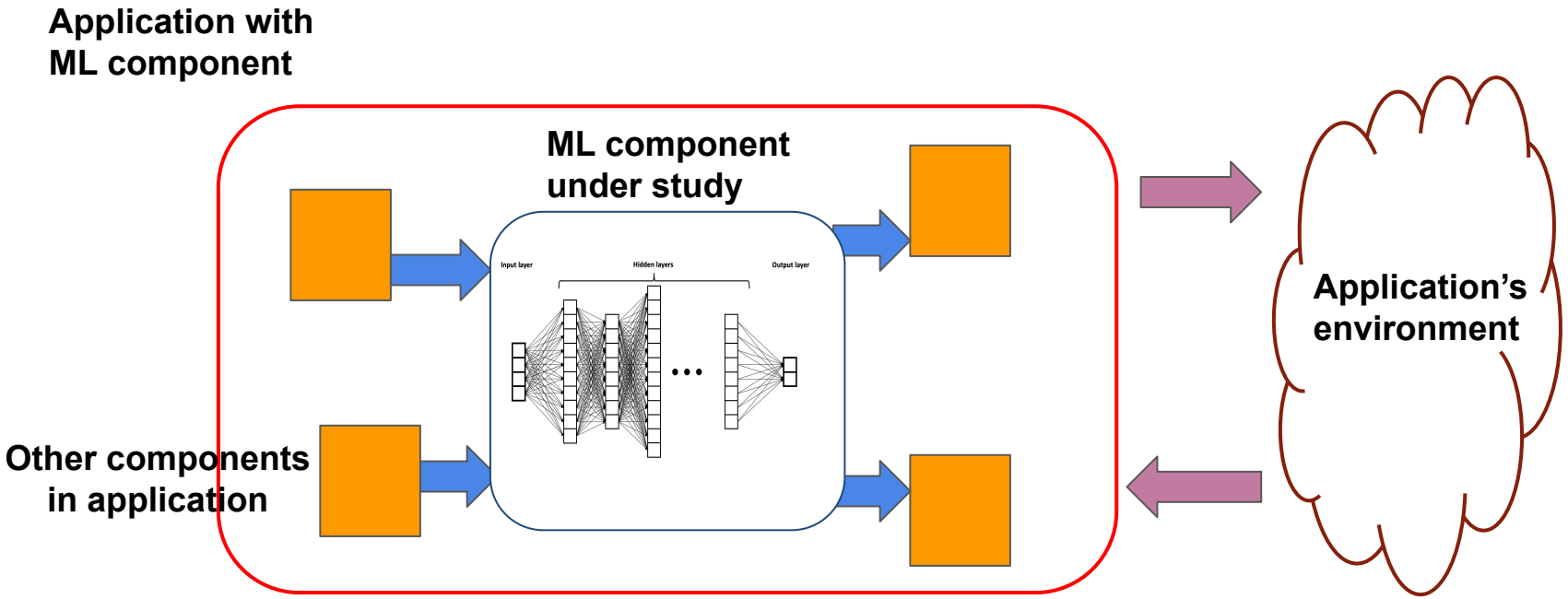
# Centrality of Machine Learning based Decisions



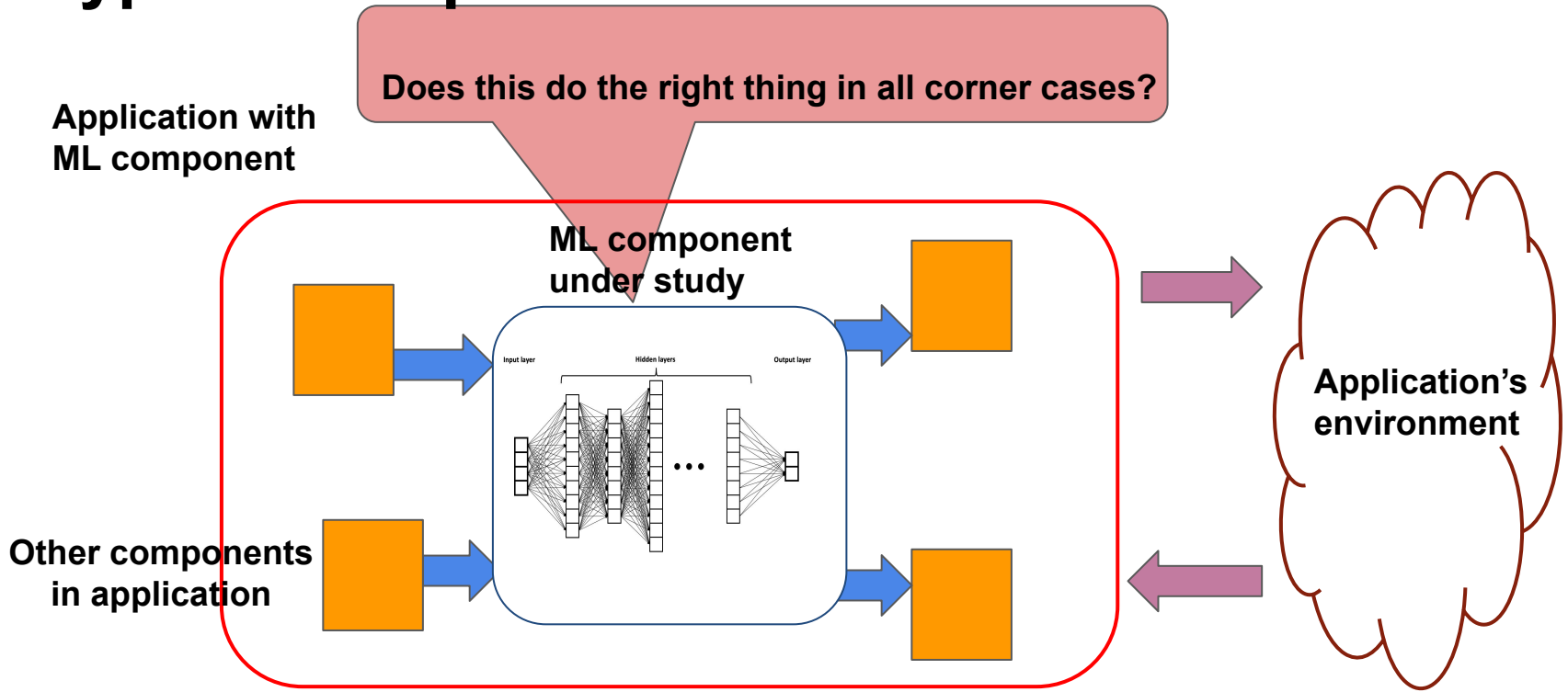

**Semi-autonomous Automatic Emergency Braking System: NSF VeHICaL project**

Source: Formal Specification for Deep Neural Networks, Seshia et al, 2018

# A Typical Setup



# A Typical Setup



# Different perspectives

- **Machine learning perspective**
  - **“Accidents”**
    - Unintended, harmful behaviour stemming from “bad” design of ML components?
    - Wrong objective function design?
    - Training based on insufficient or poorly curated data?
    - Errors due to distributional shift of inputs?

  - **Core machine learning techniques can reduce “accidents”**
    - Scalable, works in a large spectrum of real-world settings
    - **Are all corner cases covered?  Do we have proofs of correctness?**

# Different perspectives

Can we **depend** on training/designing complex networks using to **always** work **as desired** in previously unseen corner cases, when **the cost of an error is huge**?

ML based techniques to mitigate problem are **important** and **must be used**

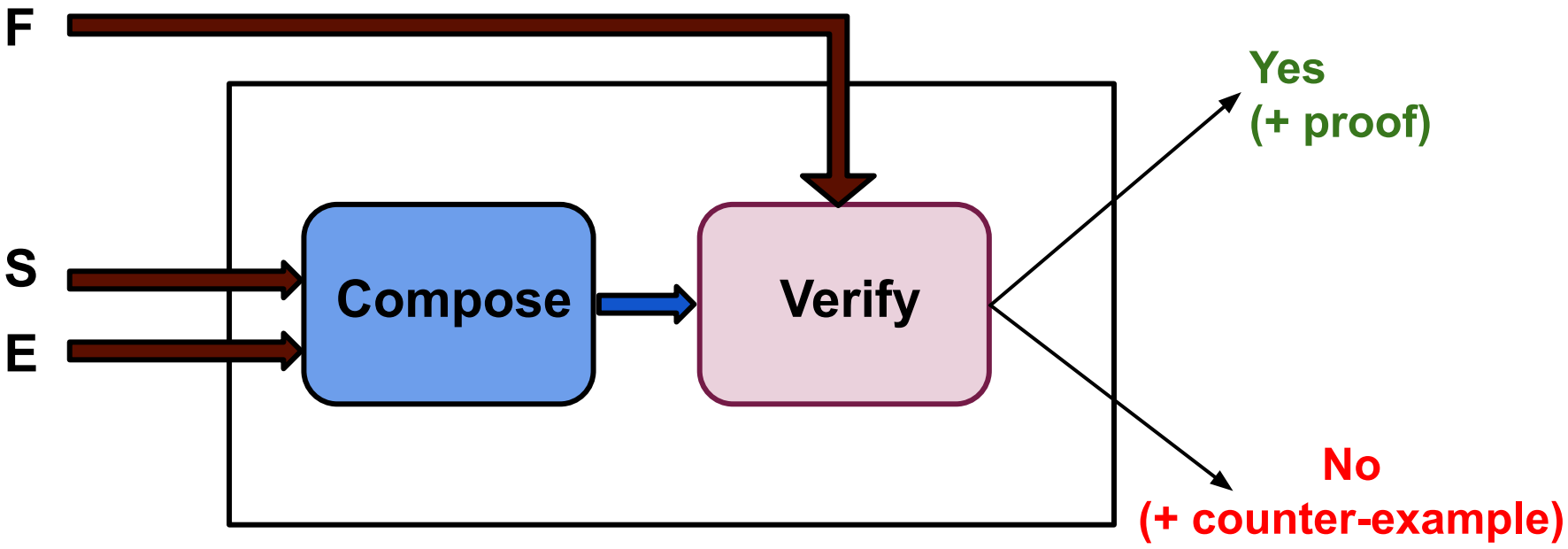
**But are these sufficient?**

# Different perspectives

- **Formal methods perspective**
  - **System:** E.g., Neural net in self-driving car
    - Mathematical model of system's behaviour (**S**)
  - **Environment:** E.g., Road, weather, traffic, driver interventions, ...
    - Mathematical model of environment's behaviour (**E**)
  - **Property:** A precise formulation (**F**) of **acceptable behaviour of S operating in E**

  - Algorithmic search of proof space
    - Either obtain a proof that system satisfies property
      - $(S \parallel E) \models F$
    - Counterexample (network inputs) that demonstrate violation of property
      - Model of $(S \parallel E) \wedge \neg F$

# Different perspectives




Ref: Towards Verified Artificial Intelligence, Seshia, Sadigh and Sastry

# Different perspectives

## Formal methods perspective

- Hugely successful in hardware industry, software industry
  - Every processor from Intel/AMD has parts of the design formally verified
  - Every time you fly an Airbus aircraft, large parts of auto-pilot software formally verified
  - Every time you insert a USB device into a Windows machine, formal verification of downloaded drivers happens

- **Can we make the technology work for AI/ML based systems?**

# Different perspectives

**FM in ML goes beyond proofs/counterexamples of safety properties**

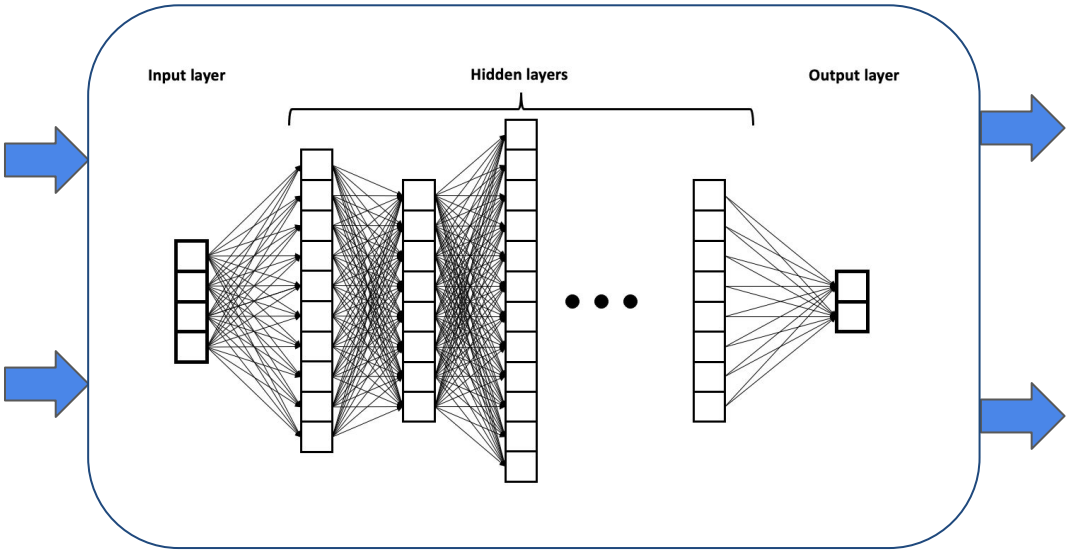
**Can we use formal methods based reasoning to**

- Verify correctness of algorithms used to train?
- Do correct-by-construction design of ML components?
- Provide explanations based on formal models?
- Fish out adversarial inputs for well-trained ML components?
- Analyze robustness, fairness, privacy, security, transparency etc.?

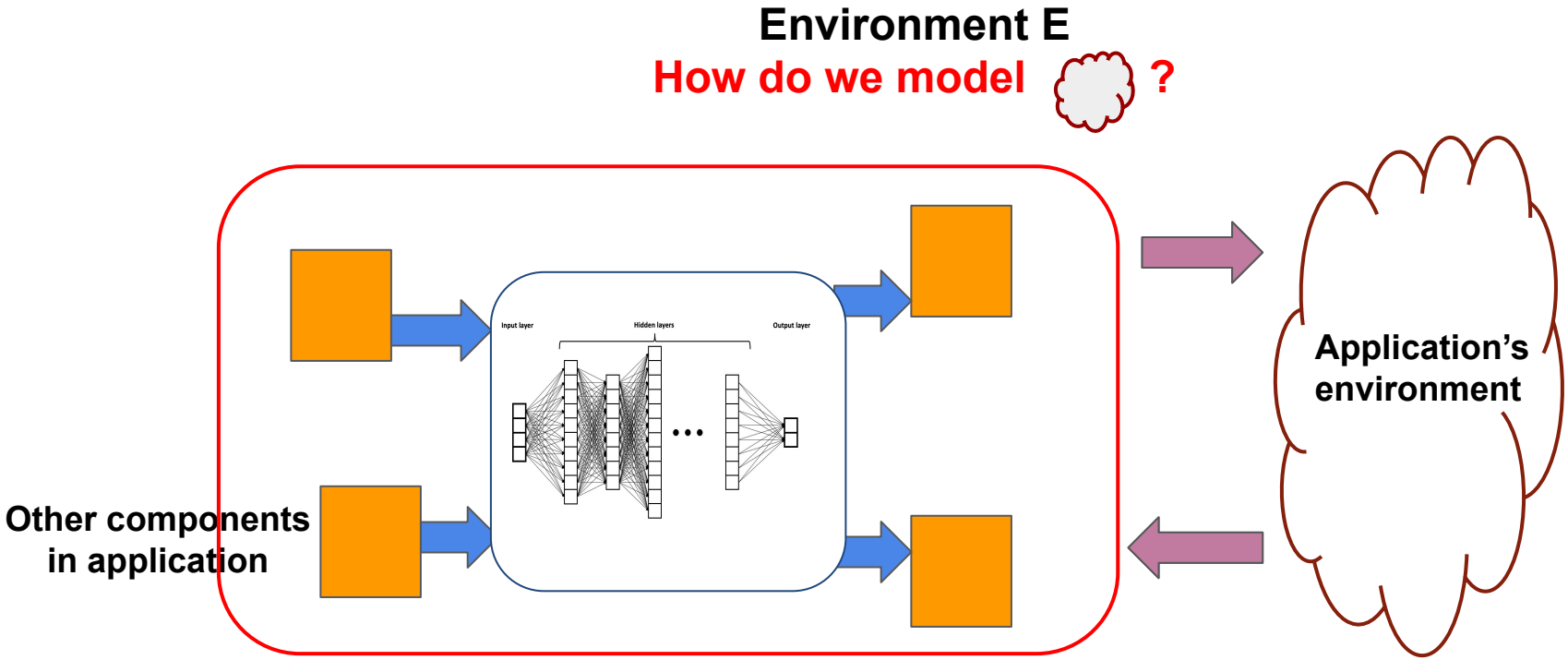
# Some common problems

**System S**

**High dimn input space, parameter space: scalability of analysis?**

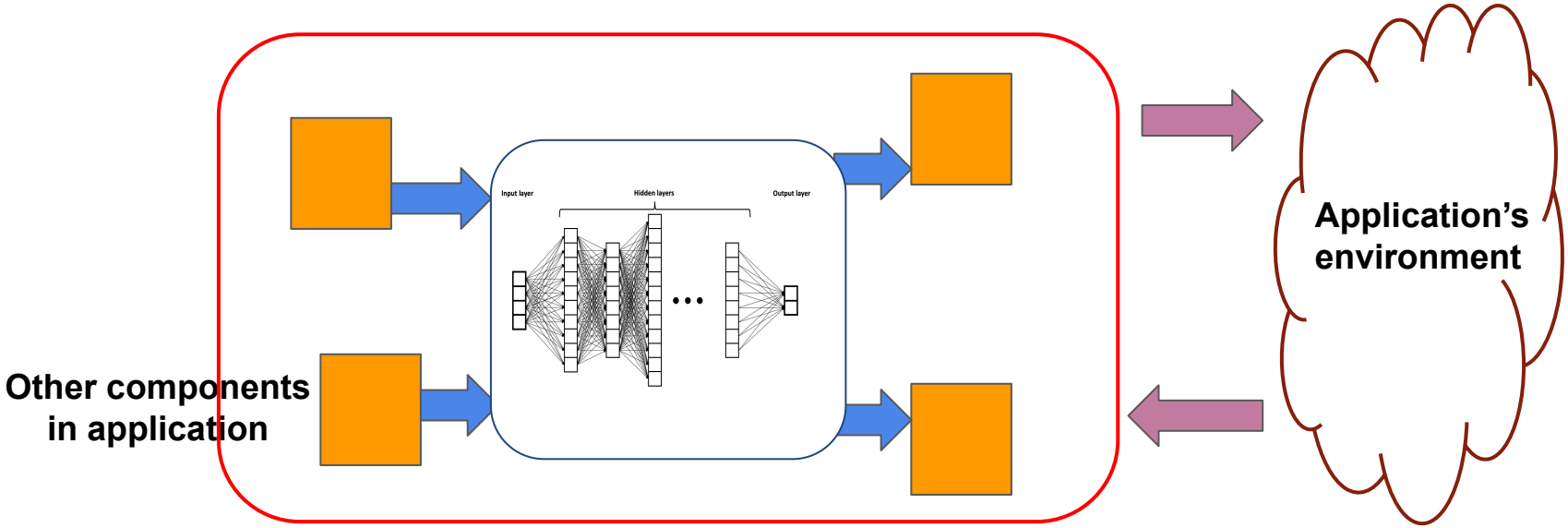


# Some common problems



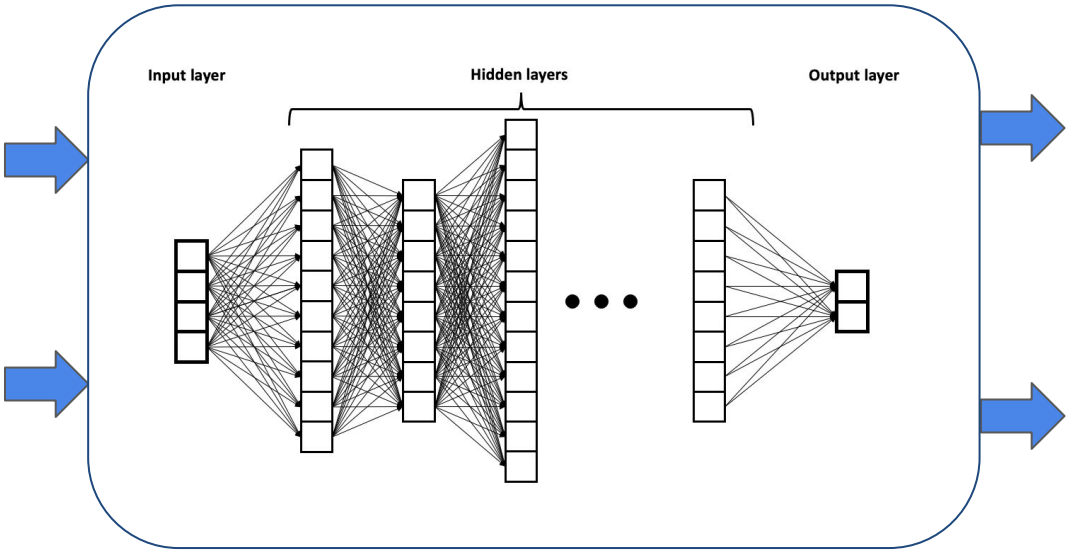
# Some common problems

**Property F: (Vehicle within 5m on left) ⟹ ¬ (Steer left)**



# Some common problems

**What is the corresponding property for S?**



# Realistic expectations

- Given scale and complexity of today's AI/ML based systems
  - Challenging, if not impossible, to design correct-by-construction ML system, or formally verify overall correct operation without restrictive/unrealistic assumptions
  - Nascent area, lots of promising ideas in literature
- Therefore,
  - **Core ML techniques, Formal Analysis/Verification AND Run-Time Assurance needed**

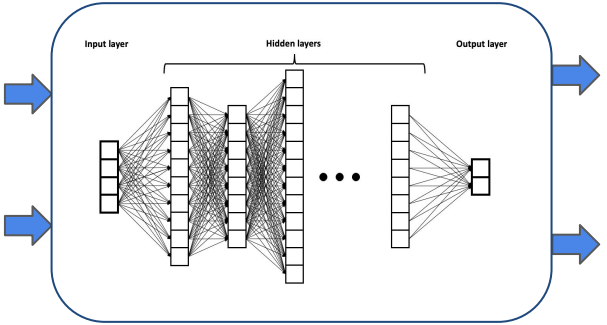
**Formal Methods and Machine Learning must help each other**

# Some Buzzing Research Topics

- Specifying properties for ML components
- Modeling environments and neural networks
- Abstract interpretation for analyzing deep neural networks
- Customized constraint solvers
- Verified Reinforcement Learning
- Robustness analysis through formal methods lens
- Explainability of ML components: logic based approach

# Some Additional Details

# Modeling the system




- Very high dimensional input space
- Need abstraction mechanisms suitable for scale of ML component complexity
  - Walking a tight rope -- computational efficiency vs precision of analysis
- Use logical formalisms to “explain” ML components
  - Some of these can be used as models
- Model systems in context
    - Perhaps not necessary to model arbitrary behaviours

# Modeling environment


- Uncertainty omnipresent:  First class entity in reasoning
- Some things are inherently hard to model
  - Human behaviour, traffic conditions
- Need to combine probabilistic and non-deterministic modeling intelligently
- Markov Decision Processes (MDPs), probabilistic programs, …
- Abstractions in environment modeling

# Specification of what is desired behaviour

- Often hard to formalize
  - Significant chunk of time spent on this even in software/hardware verification
- “Data as specification” vs “formal specification”
  - Can this gap be bridged?
  - Specification mining from behaviours, traces?
- Quantitative vs Boolean specifications
  - Quantitative specs often have an optimization flavour
  - Does a system satisfy/fail a property or get a formal score for property satisfaction?
- Run-time monitors

# Practically “efficient” computational techniques

- Hardware & software verification settings
  - Symbolic model checking, SAT/SMT solvers, numerical simulation techinques ..
- AI/ML context
  - Data generation, satisfying soft, hard, distributional constraints (realism)
  - Efficient constraint solving techniques with ReLUs, sigmoids, etc.
  - New abstraction/refinement techniques for ReLUs, sigmoids for sound analysis
  - Compositional reasoning
    - Assume-guarantee reasoning for Boolean models/specifications relatively mature
    - Similar reasoning for probablistic/quantitative models/specifications?