Machine Learning in Production

Safety

Mitigating more mistakes…

Fundamentals of Engineering AI-Enabled Systems

Holistic system view: AI and non-AI components, pipelines, stakeholders, environment interactions, feedback loops

Requirements:
System and model goals
User requirements
Environment assumptions
Quality beyond accuracy
Measurement
Risk analysis
Planning for mistakes

Architecture + design:
Modeling tradeoffs
Deployment architecture
Data science pipelines
Telemetry, monitoring
Anticipating evolution
Big data processing
Human-AI design

Quality assurance:
Model testing
Data quality
QA automation
Testing in production
Infrastructure quality
Debugging

Operations:
Continuous deployment
Contin. experimentation
Configuration mgmt.
Monitoring
Versioning
Big data
DevOps, MLOps

Teams and process: Data science vs software eng. workflows, interdisciplinary teams, collaboration points, technical debt

Responsible AI Engineering

Provenance, versioning, reproducibility

Safety

Security and privacy

Fairness

Interpretability and explainability

Transparency and trust

Ethics, governance, regulation, compliance, organizational culture

Reading

S. Mohseni et al., Practical Solutions for Machine Learning Safety in Autonomous Vehicles. SafeAI Workshop@AAAI (2020).

Learning Goals

- Understand safety concerns in traditional and AI-enabled systems
- Understand the importance of ML robustness for safety
- Apply hazard analysis to identify risks and requirements and understand their limitations
- Discuss ways to design systems to be safe against potential failures
- Suggest safety assurance strategies for a specific project
- Describe the typical processes for safety evaluations and their limitations

AI Safety

Amodei, Dario, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. "Concrete problems in AI safety." arXiv preprint arXiv:1606.06565 (2016).

Your Favorite AI Dystopia?

The AI Alignment Problem

AI is optimized for a specific objective/cost function

- Inadvertently cause undesirable effects on the environment
- e.g., Transport robot: Move a box to a specific destination
- Side effects: Scratch furniture, bump into humans, etc.,

Side effects may cause ethical/safety issues (e.g., social media optimizing for clicks, causing teen depression)

Difficult to define sensible fitness functions:

- Perform X *subject to common-sense constr. on the environment*
- Perform X *but avoid side effects to the extent possible*

Reward Hacking

PlayFun algorithm pauses the game of Tetris indefinitely to avoid losing

When about to lose a hockey game, the PlayFun algorithm exploits a bug to make one of the players on the opposing team disappear from the map, thus forcing a draw.

Self-driving car rewarded for speed learns to spin in circles

Example: Coast Runner

Reward Hacking

- AI can be good at finding loopholes to achieve a goal in unintended ways
- Technically correct, but does not follow *designer's informal intent*
- Many possible causes, incl. partially observed goals, abstract rewards, feedback loops
- In general, a very challenging problem!
  - Difficult to specify goal & reward function to avoid all possible hacks
  - Requires careful engineering and iterative reward design

AI Alignment Problem = Requirements Problem

Recall: "World vs. machine"

- Identify stakeholders in the environment & possible effects on them
- Anticipate side effects, feedback loops
- Constrain the scope of the system
- Perfect contracts usually infeasible, undesirable

But more requirements engineering unlikely to be only solution

Existential AI Risk

Existential risk and AI alignment common in research

Funding through *longtermism* branch of effective altruism
(Longtermism is the view that positively influencing the longterm future is a key moral priority of our time.)

Ord estimates 10% existential risk from unaligned AI in 100 years

Our view: AI alignment not a real concern for the kind of ML-enabled products we consider here

Ord, Toby. The precipice: Existential risk and the future of humanity. Hachette Books, 2020.

Speaker notes

Relevant for reinforcement learning and AGI

More pressing AI isks?

AI

Ethicists fire back at 'AI Pause' letter they say 'ignores the actual harms'

Devin Coldewey @techcrunch / 8:18 PM EDT • March 31, 2023

"Those hypothetical risks are the focus of a dangerous ideology called longtermism that ignores the actual harms resulting from the deployment of AI systems today," they wrote, citing worker exploitation, data theft, synthetic media that props up existing power structures and the further concentration of those power structures in fewer hands.

Practical Alignment Problems

Does the model goal align with the system goal? Does the system goal align with the user's goals?

- Profits (max. accuracy) vs fairness
- Engagement (ad sales) vs enjoyment, mental health
- Accuracy vs operating costs

Test model *and* system quality *in production*

(see requirements engineering and architecture lectures)

System Safety

Defining Safety

Prevention of a system failure or malfunction that results in:

- Death or serious injury to people
- Loss or severe damage to equipment/property
- Harm to the environment or society

Safety is a system concept

- Can't talk about software/ML being "safe"/"unsafe" on its own
- Safety is defined in terms of its effect on the **environment**

Safety != Reliability

Reliability = absence of defects, mean time between failure

Safety = prevents accidents, harms

Can build safe systems from unreliable components (e.g. redundancy, safeguards)

System may be unsafe despite reliable components (e.g. stronger gas tank causes more severe damage in incident)

Accuracy is usually about reliability!

Safety of AI-Enabled Systems

Dr. Emily Slackerman Ackerman
@EmilyEAckerman · **Follow**

i (in a wheelchair) was just trapped *on* forbes ave by one of these robots, only days after their independent roll out. i can tell that as long as they continue to operate, they are going to be a major accessibility and safety issue. [thread]

pittnews.com
Everything we know about the Starship food delivery robots
The white, 2-foot tall battery-powered delivery robots will be sharing the sidewalk with Oakland pedestrians starting sometime in late …

7:27 PM · Oct 21, 2019

3.8K **Reply** **Copy link**

Speaker notes

Systems can be unsafe in unexpected ways

Safety of AI-Enabled Systems

Speaker notes

Systems can be unsafe in unexpected ways

Safety is a broad concept

Not just physical harms/injuries to people

Includes harm to mental health

Includes polluting the environment, including noise pollution

Includes harm to society, e.g. poverty, polarization

Case Study: Self-Driving Car

How did traditional vehicles become safer?

National Traffic & Motor Safety Act (1966):

- Mandatory design changes (head rests, shatter-resistant windshields, safety belts)
- Road improvements (center lines, reflectors, guardrails)
- Significant reduction (13-46%) in traffic fatalities

Autonomous Vehicles: What's different?

Ford Taps the Brakes on the Arrival of Self-Driving Cars

HYPE CYCLE —

The hype around driverless cars came crashing down in 2018

Top Toyota expert throws cold water on the driverless car hype

- In traditional vehicles, humans ultimately responsible for safety
  - Built-in safety features (lane keeping, emergency braking)
  - i.e., safety = human control + safety mechanisms
- Use of AI in autonomous vehicles: Perception, control, routing, etc.,
  - Inductive training: No explicit requirements or design insights
  - **Can ML achieve safe design solely through lots of data?**

Demonstrating Safety

Q. More miles tested => safer?

Challenge: Edge/Unknown Cases

- Gaps in training data; ML unlikely to cover all unknown cases
- Is this a unique problem for AI? What about humans?

Safety Engineering

Safety Engineering: An engineering discipline which assures that engineered systems provide acceptable levels of safety.

Typical safety engineering process:

- Identify relevant hazards & safety requirements
- Identify potential root causes for hazards
- For each hazard, develop a mitigation strategy
- Provide evidence that mitigations are properly implemented

Improving Safety of ML-Enabled Systems

Anticipate problems (hazard analysis, FTA, FMEA, HAZOP, ...)

Anticipate the existence of unanticipated problems

Plan for mistakes, design mitigations (recall earlier lecture!)

- Human in the loop
- Undoable actions, failsoft
- Guardrails
- Mistaked detection
- Redundancy, ...

Demonstrating and Documenting Safety

Demonstrating Safety

Two main strategies:

1. **Evidence of safe behavior in the field**
    - Extensive field trials
    - Usually expensive
2. **Evidence of responsible (safety) engineering process**
    - Process with hazard analysis, testing mitigations, etc
    - Not sufficient to assure safety

Most standards require both

Demonstrating Safety

How do we demonstrate to a third-party that our system is safe?

Safety & Certification Standards

- Guidelines & recommendations for achieving an acceptable level of safety
- Examples: DO-178C (airborne systems), ISO 26262 (automotive), IEC 62304 (medical software), Common Criteria (security)
- Typically, **prescriptive & process-oriented**
  - Recommends use of certain development processes
  - Requirements specification, design, hazard analysis, testing, verification, configuration management, etc.,
- Limitations
  - Most not designed to handle ML systems (exception: UL 4600)
  - Costly to satisfy & certify, but effectiveness unclear (e.g., many FDA-certified products recalled due to safety incidents)
- Good processes are important, but not sufficient; provides only indirect evidence for system safety

Certification Standards: Example

Demonstrating Safety

Two main strategies:

1. **Evidence of safe behavior in the field**
    - Extensive field trials
    - Usually expensive
2. **Evidence of responsible (safety) engineering process**
    - Process with hazard analysis, testing mitigations, etc
    - Not sufficient to assure safety

Most standards require both, but often not sufficient!

Assurance (Safety) Cases

- An emerging approach to demonstrating safety
- An explicit argument that a system achieves a desired safety requirement, along with supporting evidence
- Structure:
  - Argument: A top-level claim decomposed into multiple sub-claims
  - Evidence: Testing, software analysis, formal verification, inspection, expert opinions, design mechanisms…

Documenting Safety with Assurance (Safety) Cases

Assurance Cases: Example

Questions to think about:

- Do sub-claims imply the parent claim?
- Am I missing any sub-claims?
- Is the evidence strong enough to discharge a leaf claim?

Assurance Cases: Example

Aurora Safety Case

Discussion: Assurance Case for Recommender

Assurance Cases: Benefits & Limitations

- Provides an explicit structure to the safety argument
  - Easier to navigate, inspect, and refute for third-party auditors
  - Provides traceability between system-level claims & low-level evidence
  - Can also be used for other types of system quality (security, reliability, etc.,)
- Challenges and pitfalls
  - Informal links between claims & evidence, e.g., Does the sub-claims actually imply the top-level claim?
  - Effort in constructing the case & evidence: How much evidence is enough?
  - System evolution: If system changes, must reproduce the case & evidence
- Tools for building & analyzing safety cases available
  - e.g., ASCE/GSN from Adelard
  - But ultimately, can't replace domain knowledge & critical thinking

Testing is still important!

Other Challenges

- Safe Exploration
  - Exploratory actions "in production" may have consequences
  - e.g., trap robots, crash drones
- Robustness to Drift
  - Drift may lead to poor performance that may not even be recognized
- Scalable Oversight
  - Cannot provide human oversight over every action (or label all possible training data)
  - Use indirect proxies in telemetry to assess success/satisfaction

Robustness for ML-based Systems

Robustness

Environment sometimes **deviates** from expected, normal conditions

- Extreme weathers, unexpected obstacles, etc.,
- Erratic user behaviors; unusually high service demand
- Adversarial actors; users trying to game your system, etc.,

Does your system work reasonably well under these deviations? i.e., is it *robust*?

Most safety-critical systems require some level of robustness

- Not enough to show that system is safe in normal conditions