Developing and Operating AI/ML Capabilities at the Edge

2022 DoD AI/ML TEM Special Session

Kevin Pitstick, CMU/SEI
Benji Maruyama, AFRL
Jeffrey Chrabaszcz, CMU/SEI

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM22-0948

What does "using AI/ML at the edge" mean?

- *Common thought:* Developing models in the cloud and then deploying them on edge devices.
- Reality is more complex:
  - AI/ML engineering stages include:
    - data collection, data curation, model training, model testing, model deployment, model inferencing, etc.
  - Cloud-to-edge continuum: there is not just "one" edge
- Other examples:
  - Set up "near edge" nodes (fog computing) for retraining, deploy back to "far edge" nodes
  - Federated learning: training models across multiple edge devices that each utilize their locally collected data

What are challenges for AI/ML at the edge?

- Good AI/ML models are resource-intensive, and edge devices are resource-constrained
- AI/ML models need lots of data, but it's hard to keep all the data at the edge
- Edge AI/ML models usually aren't trained on their deployment platforms
- Collecting data at the edge can easily result in garbage training data
- AI/ML algorithms can be easy to fool
- AI/ML algorithms are only good for the data they were trained on
- Scaling AI/ML at the edge leads to increased compatibility and interoperability complexity
- Human-machine teaming at the edge requires accommodating user restrictions

Good AI/ML models are resource-intensive, and edge devices are resource-constrained

Challenge

- Size, weight, and power (SWaP) limitations at the edge
- Requires trade-offs between accuracy and speed

Solution

- Model compression and acceleration techniques:
  - Network pruning
  - Knowledge distillation
  - Quantization (16-bit, 8-bit)
  - Hardware-optimized inference engines (NVIDIA TensorRT, Intel OpenVINO)

AI/ML models need lots of data, but it's hard to keep all the data at the edge

Challenge

- Limited storage & network connectivity at the edge

Solution

- Data summarization: large amount of data compressed into a smaller amount of data
- Data filtering: keep only a subset of the data & throw away the rest
- Simpler ML models can be used to help with summarizing & filtering at the edge
- Federated learning

Edge AI/ML models usually aren't trained on their deployment platforms

Challenge

- Models are generally trained on high-compute servers.
- When deployed to the edge, models may not run or run slower than expected.

Solution

- Good software engineering design during model development
  - Identify training/deployment platforms early, use proof of concept as needed

Collecting data at the edge can easily result in garbage training data

Challenge

- Due to policy restrictions, data collected at the edge may require cleaning to remove PII (personally identifiable information).
  - If done improperly, this can destroy vital information in the samples.
- Some data is only useful for training if metadata about the environmental conditions is also recorded.
- In extreme cases (privacy or networking), data can't be stored or transmitted.

Solution

- Define and follow process for data collection and cleaning
- Understand environmental conditions of operation
- Federated learning: retrain at the edge if data cannot be stored or transmitted

AI/ML algorithms can be easy to fool

Challenge

- The edge is often contested. Adversaries will attempt to influence decision-making of algorithms.
  - e.g., direct control, input data manipulation, training data manipulation (data poison)

Solution

- Adversarial machine learning: investigates these attacks and strategies for making robust AI/ML systems
  - Potential tradeoff: reduced accuracy during normal operation

AI/ML algorithms are only good for the data they were trained on

Challenge

- AI/ML model accuracy is dependent on training data.
- Operational data at the edge is constantly changing, due to uncertainty and drift.

Solution

- Match training data to operational data as best as possible.
- Domain adaptation: techniques to allow models trained in one domain to perform well in other domains

Scaling AI/ML at the edge leads to increased compatibility and interoperability complexity

Challenge

- The future contains edge deployments of 100s/1000s of heterogeneous, specialized devices that need to collaborate.
- Compatibility: models and training/inference software needs to be specialized.
- Interoperability: devices with specialized roles need to coordinate for common goal

Solution

- Disciplined software systems architecture practices
- Innovations in the software stack for edge AI/ML

Human-machine teaming at the edge requires accommodating user restrictions

Challenge

- Users at the edge have low attention due to being in uncertain, rapidly changing, and high pressure situations.
- However, human oversight is often required required of AI/ML output.

Solution

- Human factors and ergonomics need to be considered in the design process.
- Advances in trust and autonomy: how can we gain trust in our AI/ML algorithms in order to allow them to act with autonomy?