Integration of Machine Learning with ns-3: Challenges and Opportunities

Eduardo Nuno Almeida
OUTLINE

Machine Learning Background

Integration of Machine Learning with ns-3

Train Machine Learning Models using ns-3

Conclusions
MACHINE LEARNING CATEGORIES

SUPERVISED LEARNING

UNSUPERVISED LEARNING

REINFORCEMENT LEARNING
SUPERVISED LEARNING

LEARN FUNCTION MAPPING FEATURES (INPUT X) TO LABELS (OUTPUT Y)

REGRESSION ALGORITHMS

Estimate Y (Output) for X (Input)

CLASSIFICATION ALGORITHMS

Classify Data into Finite Categories
REINFORCEMENT LEARNING

TRAIN AGENT TO LEARN OPTIMAL POLICY TO MAXIMIZE EPISODE’S CUMULATIVE REWARD

▪ Policy: action to take for a given state
▪ Maximize episode’s cumulative reward
▪ Episode: State → Action → Reward

✓ Learn and adapt to scenario dynamics
✓ Real-time network performance metrics
✓ Learn from experience

× Learning requires many episodes
× Requires realistic interactive environment
  – Challenging to train agents in testbeds
  – ns-3 can serve as environment
OPENAI GYM

- Standard Python API for Reinforcement Learning
- Manage interaction between RL agent and environment
- Independent of agent’s implementation
  - Allows fair and easy comparison between RL algorithms
INTEGRATION OF ML WITH NS-3

INTEGRATION WITH ML FRAMEWORKS
- Integration via third-party modules
  - ns3-gym, ns3-ai

✔ Integration with existing ML frameworks in Python
✔ Reuse existing ML models
✘ Computational performance overhead
  - Due to data exchange between processes

NATIVE INTEGRATION
- No native integrations
  - E.g., ONNX framework

✔ Improved computational performance
  - No overhead due to data exchange
✘ Additional dependency to manage
✘ Tight coupling of ns-3 and ML code
  - Code recompilation for ML model updates
  - ONNX separates runtime from ML model
NS3-GYM MODULE

- Development of OpenAI Gym RL environments in ns-3
  - Execute actions, provide observations and reward in underlying ns-3 simulation
  - Data exchanged via protobuf messages over ZMQ / sockets

NS3-GYM MODULE ANALYSIS

ADVANTAGES

▪ Seamless integration with OpenAI Gym
▪ Helper scripts to launch ns-3 and RL agent
▪ Examples provided by the module
▪ Community on GitHub and ns-3-users

CHALLENGES

▪ Multiple issues reported in GitHub
  – No support for matrix values
  – No reshaping of Box container
  – No check if values are within defined range
▪ Rare updates to module
▪ Documentation only available in the paper
  – No quick-start guides or tutorials
▪ Computational overhead due to sockets
NS3-AI MODULE

- Integration with existing Python ML frameworks
  - API to read and write data between ns-3 and ML process
  - Data exchange via shared memory

NS3-AI MODULE ANALYSIS

ADVANTAGES

- **Easy** integration with ML frameworks and ns-3
- **Flexible** and **powerful** data exchange mechanism
  - Can be extended beyond AI applications
- Ongoing **GSoC 2023** to improve ns3-ai
  - OpenAI Gym interface, performance, ...
- Good **documentation** and examples
- Community on GitHub and ns-3-users

CHALLENGES

- Ongoing fixes / improvements to main issues
- No integration with applications other than **Python**
- No **helper scripts** to launch ns-3 and ML application

NS3-AI ADDITIONAL USE CASES

- Shared memory mechanism can be used in scenarios beyond AI
- Enables integration with any external Python application
  - Optimization solvers
  - Real applications (e.g., network controllers)
INTEGRATION OF SOLVERS WITH NS3-AI

OFFLINE METHODOLOGY

INPUT DATA → EXTERNAL PYTHON SOLVER → FILE WITH SOLVER OUTPUT → RECREATE SCENARIO FROM FILE → ns-3 → RESULTS

No Interaction between Solver and ns-3.
Create Offline Simulation Replicating Solver Output.

ONLINE METHODOLOGY

EXTERNAL PYTHON SOLVER

INPUT DATA → ns-3

Real-time Interaction between Solver and ns-3.
Dynamic Simulations Based on Solver Output.
ONNX FRAMEWORK

- **Open Neural Network Exchange (ONNX)**
  - Open format to *represent* ML models
  - *Portable* and *interoperable* among platforms and frameworks

- **Use cases**
  - Build and share of ML models
  - Deployment of ML models for *inference* using ONNX runtime
TRAINING ML MODELS USING NS-3

GENERATE DATASETS FOR SUPERVISED LEARNING

- When experimental data not available
- When insufficient experimental data
- Augment / transform existing experimental dataset
  - Collect results for different scenario parameters

CREATE REALISTIC RL ENVIRONMENTS

- Realistic interactive environment for RL
- Train RL agents with offline learning
- Pre-train / improve policies for online learning
- Evaluate and compare RL trained policies
TRAINING ML MODELS USING NS-3

CHALLENGES

▪ Existing models in ns-3 may not fully capture environment dynamics
  – Extreme scenarios
▪ Non-existent models
▪ Generate realistic datasets with randomness and noise
▪ Computational performance

OPPORTUNITIES

▪ Improve ns-3 models with trace-based or ML
  – Collect experimental data in testbed
  – Accurate and customized models
  – Specific to scenario
▪ Trace-based simulation approaches
  – Accurate, repeatable and reproducible
  – Propagation loss, channel occupancy, rate adaptation, MIMO, ...
▪ ML-based models
  – ML Propagation Loss (MLPL) model
TRACE-BASED SIMULATION APPROACH

NETWORK TRACES COLLECTION

SCENARIO PARAMETERS

RUN TESTBED EXPERIMENTS

EXPERIMENTAL NETWORK TRACES

REPEAT AND REPRODUCE EXACT EXPERIMENTAL CONDITIONS IN NS-3

NETWORK TRACES USAGE

EXPERIMENTAL NETWORK TRACES

LOAD NETWORK TRACES

RUN NS-3 SIMULATION

NS-3 RESULTS

INTEGRATION OF MACHINE LEARNING WITH NS-3: CHALLENGES AND OPPORTUNITIES

Eduardo Nuno Almeida | Workshop on ns-3 (WNS3) 2023

**ML PROPAGATION LOSS (MLPL) MODEL**

**TRAINING**

- EXPERIMENTAL PROPAGATION LOSS
- TRAIN PROPAGATION LOSS REGRESSION
- PROPAGATION LOSS FUNCTION

**ESTIMATION**

- TX / RX NODES COORDINATES
- PROPAGATION LOSS FUNCTION
- NS-3 SIMULATION
- PROPAGATION LOSS VALUES

TRAINING ML MODELS USING NS-3

ML QUALITY OF SERVICE ESTIMATOR

- Estimate QoS based on
  - Users traffic demand
  - UAV positions
- Convolutional neural network
- One estimator per QoS metric
  - Throughput, Delay, PLR
- **Dataset** generated in ns-3

---

TRAINING ML MODELS USING NS-3

DRL TRAFFIC-AWARE UAV PLACEMENT

- Position FAP
- According to users traffic demand
- Maximize network utility
- Trained and evaluated with ns3-gym

INTEGRATION OF MACHINE LEARNING WITH NS-3: CHALLENGES AND OPPORTUNITIES

Eduardo Nuno Almeida | Workshop on ns-3 (WNS3) 2023

TRAINING ML MODELS USING NS-3
DRL DATA-DRIVEN WI-FI RATE ADAPTATION

- Modulation and Coding Scheme (MCS)
- According to channel state
- Maximize throughput and Frame Success Ratio (FSR)
- Trained and evaluated with ns3-gym + trace-based

\[
R = \frac{\text{MCS}_n}{\text{MCS}_7} \times \text{FSR}, \quad n \in \{1, \ldots, 7\}
\]

CONCLUSIONS

- Integration of external ML frameworks via ns3-ai and ns3-gym
  - Opportunity to improve the modules
  - Consider supporting ONNX for deployment of ML models

- ns3-ai powerful tool for applications beyond AI
  - Integration with Python applications (e.g., solvers or controllers)

- ns-3 interesting tool to train and evaluate ML models
  - Generate training datasets for supervised learning
  - Create realistic interactive environments for RL
  - Can be enhanced with trace-based or ML-based models
QUESTIONS?

Integration of Machine Learning with ns-3: Challenges and Opportunities

Eduardo Nuno Almeida | INESC TEC & FEUP, Portugal
eduardo.n.almeida@inesctec.pt