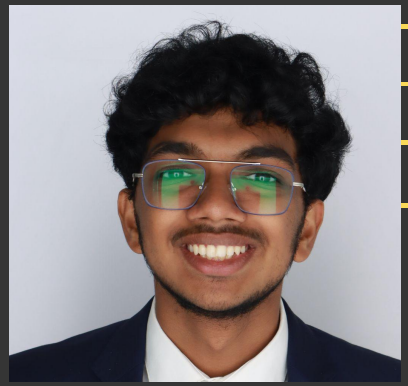
ARM

NNAS for RF Modulation Classification

Ashwin Menon

Team Members



Ashwin Ranjith Menon
Team Member

Contact:
menonar@mail.uc.edu

Dr. Boyang Wang
Faculty Advisor

Contact:
wang2ba@ucmail.uc.edu

Project Abstract

With this project, we're exploring reinforcement learning to perform neural network architecture search which will improve radio frequency modulation classification on constrained hardware. Deep learning models for RF modulation deliver strong accuracy but require high performance devices, making them impractical for embedded and edge devices. We will be using the Q-learning approach to generate neural networks with small parameter counts that preserve accuracy

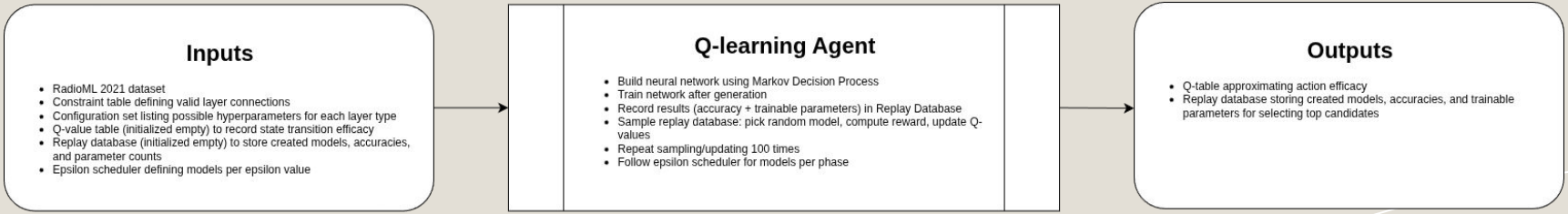
User Stories

1. As an algorithm developer, I want to implement an automated process that can perform architecture search for the purposes of modulation classification to accelerate the discovery of robust models for wireless communication systems

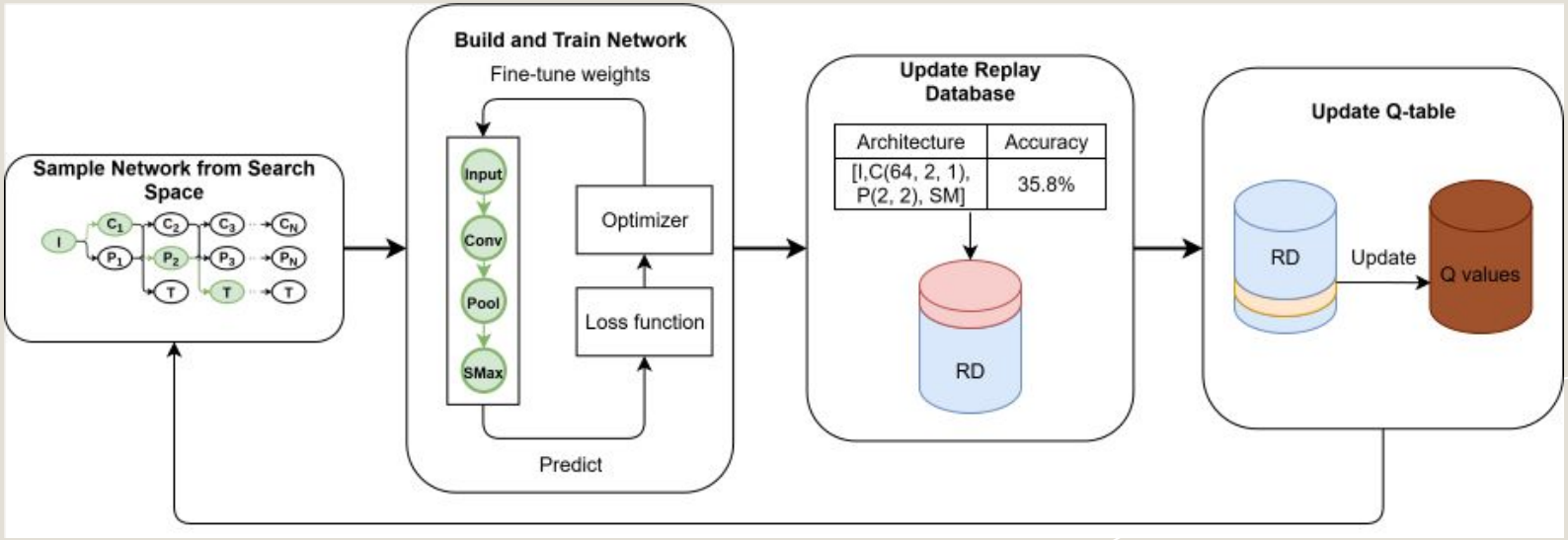
2. As an algorithm developer, I want to use reinforcement learning to explore novel model topologies so that I don't have to manually build and test configurations through trial and error

3. As an algorithm developer, I want to leverage I/Q signal data provided by RadioML to identify the most effective model for classifying modulation types

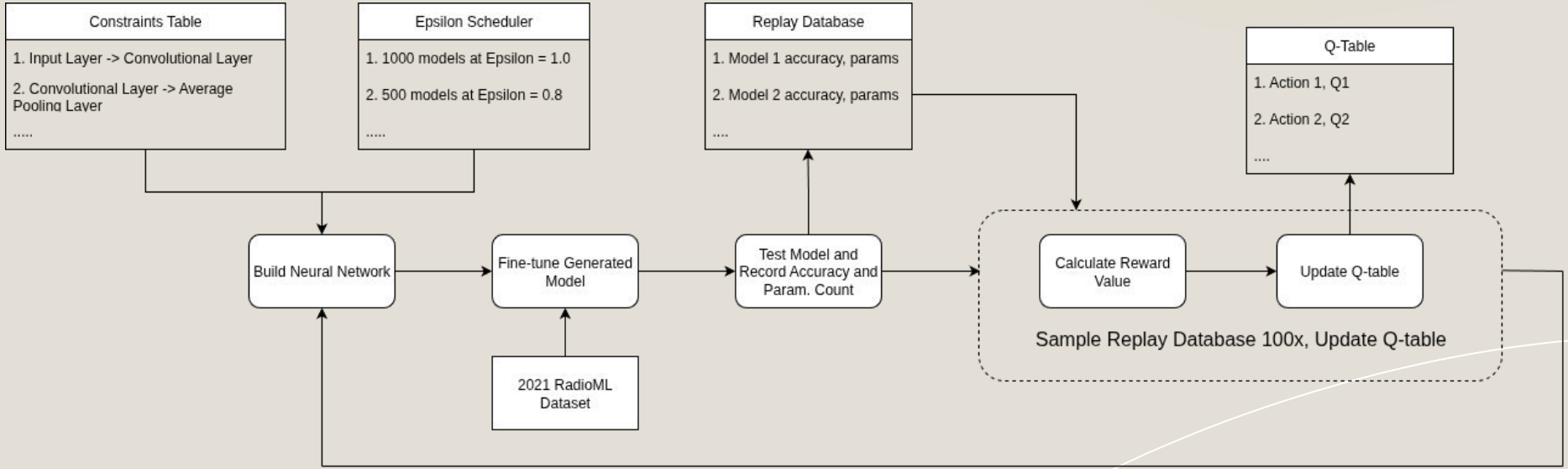
Design Diagram - Level 0



Design Diagram - Level 1



Design Diagram - Level 2



Project Constraints

1

Economic

- Used open-source frameworks and datasets
- Used university provided compute resources

2

Social

- Seeks to benefit the public by designing neural networks for cheap edge devices
- Brings the benefits of machine learning to the public

3

Professional

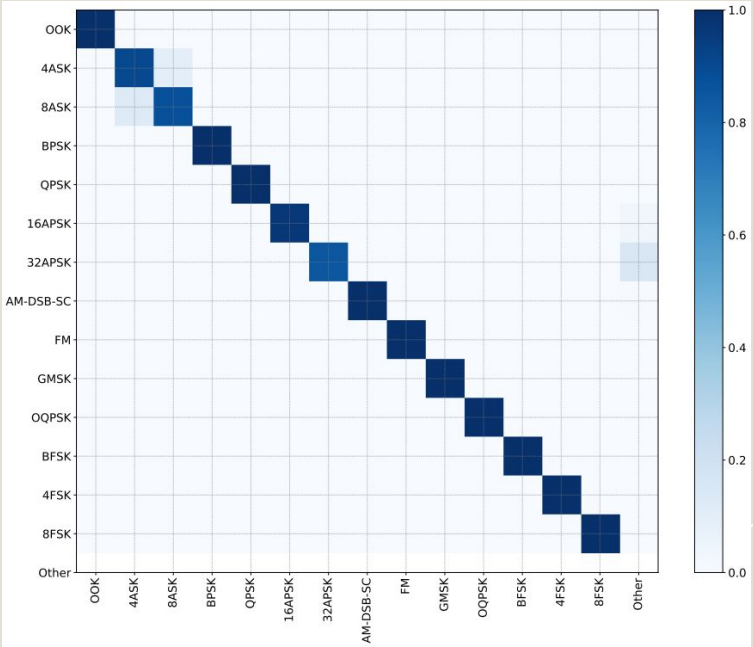
- Requires specialized expertise in conducting scientific research
- Also requires effort to review existing literature

Current Project Status

TABLE IV: Performance of Neural Networks for RF Classifications

Model	Model Score	No. of Parameters	Parameter Reduction Rate	Size (KB)	Acc. (≥6 dB SNRs)	Acc. at 30 dB SNR	Subset Acc. (≥6 dB SNRs)	Subset Acc. at 30 dB SNR
Baseline	N/A	717,252	N/A	5,950	97.76%	98.69%	99.94%	99.96%
Pruned Baseline ($h = 2$)	N/A	13,981	98.20%	280	84.38%	85.33%	99.26%	99.75%
Model 312	4.718×10^{-3}	**11,859**	**98.37%**	**255**	**72.49%**	**73.12%**	**96.90%**	**97.53%**
Model 1802	4.695×10^{-3}	6,371	99.13%	135	68.09%	68.50%	96.59%	97.39%
Model 2171	4.638×10^{-3}	5,379	99.26%	139	70.74%	71.20%	95.89%	97.69%
Model 1452	4.423×10^{-3}	9,179	98.74%	192	68.99%	69.18%	96.27%	96.74%
Model 2482	4.406×10^{-3}	5,995	99.26%	184	69.63%	69.88%	95.48%	95.96%
Model 1004	4.326×10^{-3}	2,335	99.69%	104	68.29%	68.05%	95.89%	96.10%
Model 2609	4.298×10^{-3}	12,919	98.23%	249	69.07%	68.49%	96.01%	95.97%
Model 859	4.276×10^{-3}	10,555	98.56%	208	69.55%	69.52%	96.60%	97.04%
Model 1684	4.168×10^{-3}	1,643	99.78%	103	66.09%	66.33%	95.49%	96.17%
Model 2437	4.137×10^{-3}	5,171	99.26%	159	67.25%	66.83%	96.14%	96.66%

Current Project Status - Contd.



Goals for the end of this term

1. Look into analysis tools and perform additional analysis if necessary to extract additional insights
2. Convert the experiment to a pipelined process for easy reproducibility
3. Clean up the project repository if necessary

Division of Work

- Analyze background research paper for MetaQNN (Ashwin)
- Analyze dataset to understand and unpack it (Ashwin)
- Smoke test RL implementation to see if replay database and Q-table are being initialized properly along with saved models (Ashwin)
- Create scoring function to rank candidate models so that we can pick top-k models from the replay database (Ashwin)
- Perform cleaning up and reorganization of code if necessary. Additionally, create pipelines for reproducibility (Ashwin)

Expected Demo at Expo

1. Dashboard with analysis of the learner’s output
2. Any additional insights to the experiment output