Assignment #4 - Weekly Senior Design Checkpoint - Ashwin Menon

User Stories

- 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
- 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
- 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

Diagrams' Overview

Level 0

The Level 0 diagram goes over the basic I/O structure of the algorithm and a very high level overview of its process. As we can see here, the Q-learning algorithm depends on a pre-defined constraint table that constrains what layer-to-layer transitions we can have when generating a neural network, which is needed for the learning process. Additionally, we can also gauge the fact that we depend on the 2021 RadioML dataset as an input to the learning algorithm. We also provide configurations that specify the hyperparameters each layer in a generated neural network can possess. Lastly, we provide the algorithm with an empty replay buffer and an empty Q-table so that it can record experiences and track layer-to-layer transition efficacy. The algorithm then learns from generated neural networks, updates the replay database and updates the Q-table multiple times (based on the given epsilon scheduler) and finally creates a comprehensive list of models generated along with their test accuracy and parameter count AND a comprehensive list of actions and their corresponding Q-values.

Level 1

The Level 1 diagram goes into slightly further detail as far as the process of learning iterations goes. We can clearly see that the model generates a neural network by sampling it from the search space (which is provided by the constraints in the constraint table combined with the configurations file that specifies the hyperparameters that can go with each layer type), the model is then finetuned (based on the 2021 RadioML dataset). Then the model's test performance is stored in the replay database, after which it is sampled multiple times. Each time it is sampled, we use the sampled information to update the Q-values in the Q-table. All these steps put together encapsulates a "learning iteration". After following the epsilon scheduler to perform the learning iterations a specified number of times for each epsilon value, the experiment is complete.

Level 2

The Level 2 diagram builds on the Level 1 diagram. Here we can clearly see where each stage of the learning iteration depends on the different inputs. We can see that the generation of the neural network depends on the epsilon scheduler and the constraints table while the fine-tuning step depends on the provided dataset. We can also see that after testing the model, we update the replay database and that the calculation of the reward value depends on the values chosen from the replay database. We can also see the connection between calculating the reward value and updating the Q-table. Finally we can observe how this process is repetitive.