Intro to AI – Project 3

Mak575 – Mihir Kulkarni Sb2756 – Shaivi Bansal

I. Introduction

The goal of this project is to develop a Machine Learning model capable of predicting the behaviour of a bot navigating a grid-based environment to catch a rat. This is built on top of a previous project developed by us where the bot was tasked with narrowing down the position and catching the rat using inference and logic. The neural network we aim to develop estimates the number of steps required for said bot to reach its target based on the current state of the environment.

In order to feed our model input data, we simulated scenarios of our previous project. The neural network leverages spatial data from the grid and probabilistic knowledge about the rat's location to make predictions.

II. Data Collection and Representation

Input Data Representation

- Bot Probability Grid (30x30): A 2D grid representing the bot's belief about its own position based on its internal knowledge and observations.
- Rat Probability Grid (30x30): A 2D grid representing the bot's belief about the rat's position based on sensing data.

Note that during the localization phase of project 2, there are no sensing actions taken by the bot. Therefore, the rat probability grid remains constant with all the open cells having equal probability of the rat being in them. The constant state of the grid is true in the case of the bot probability grid as well during the second rat catching phase, since will be no uncertainty regarding the position of the bot.

Scalar Inputs

Additional features provide context about the simulation state:

1. Remaining time steps.
2. Blocked-to-open cell ratio.
3. Current time step.
4. Manhattan distance to the target.
5. Probability at the target cell.
6. Probability of the most probable cell.

Output Data Representation:

- Target: A single scalar value representing the predicted number of steps needed for the bot to catch the rat.

Data Collection Process

Simulations on a fixed 30x30 grid were conducted with randomized placements of the bot and the rat.

III. Neural Network Architecture

The neural network uses a hybrid approach combining CNNs and fully connected layers to process spatial and scalar features.

Components:

1. **CNN layer for Grids**:
   - Two separate CNN paths processing the grids independently
   - Each part has 2 convolutional layers, ReLU activation for non-linearity
   - Max pooling layers for downsampling
   - Flattening to prepare features for dense layers
2. **Dense layer for Scalars:**
   - Fully connected layers to process scalar inputs.
   - ReLU for non-linearity
3. **Combined Layer**
   - Concatenates outputs from CNN and scalar layers
   - Dense layers reduce dimensionality and makes predictions
   - Dropout is used to prevent overfitting

IV. Training Methodology

Training Process:

- Loss Function: Mean Squared Error (MSE) to minimize prediction errors.
- Optimizer: Adam optimizer for adaptive learning rates.
- Batch Size: 32 samples per batch for stable updates.
- Learning Rate: 0.0001 to prevent overshooting during updates.
- Epochs: 20 iterations for optimal training.

V. Observations and Challenges

Observations:

- Training and validation losses reduced consistently, demonstrating effective learning
- Overfitting was avoided using dropout layers and small learning rates

Challenges:

- Predictions were not always highly accurate
- Model struggled to generalize well to completely unseen configurations despite pre-processing

VI. Conclusion

This project demonstrates the application of CNNs to spatial and probabilistic data for predictive modeling. The network effectively combines spatial features from probability grids and contextual information from scalar inputs to estimate steps needed for navigation tasks.

While the results show steady improvements in prediction accuracy, further enhancements—such as hyperparameter tuning, deeper architectures, and additional input features—can help refine the performance. This framework lays a foundation for future extensions in autonomous navigation and path optimization tasks.