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 with Batch Normalization for stability.
    - Max pooling layers for downsampling and dropout is applied for regularization
    - Flattening to prepare features for dense layers
2. **Dense layer for Scalars:**
    - Fully connected layers to process scalar inputs.
    - ReLU for non-linearity
    - Dropout for regularization
3. **Combined Layer**
    - Concatenates outputs from CNN and scalar layers
    - Fully connected layers followed with dense layers
    - Final layer outputs a single scalar value representing the prediction

IV. Training Methodology

Training Process:

- Loss Function: Used Smooth L1 Loss since it is good for outliers in data
- Optimizer: Adam optimizer with weight decay (1e-5). This is for regularization
- Batch Size: 64 samples per batch for stability and efficiency
- Learning Rate: Utilized OneCycleLR for dynamic learning rate adjustments
- Epochs: 50 iterations for comprehensive training.

- Gradient Clipping: Implemented this to avoid exploding gradients, which is a situation where gradients become excessively large causing divergence in the model. In deep neural networks, gradients are computed through backpropogation through layers, which can cause gradients to get exponentially large or small values. Gradient clipping limits the magnitude of the gradients to a threshold and scales down them if needed.

V. Observations and Challenges

Observations:

- Validation losses consistently reduced during training, demonstrating effective learning.
- The model achieved excellent performance on previously unseen data with the following metrics:
    1. MAE – 6.0799
    2. R^2 - 0.9515
- Dynamic learning rates, dropout, and batch normalization significantly improved generalization to new datasets.

Challenges:

- Earlier versions struggled with generalization, but issues were resolved with:
    1. Normalization of scalar features.
    2. Batch normalization and gradient clipping for stability.

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.

Techniques like dropout, batch normalization, and dynamic learning rates improved generalization. Testing on new data confirmed the model's ability to generalize.