Checkpoint Report: Reinforcement Learning

Team - 57

Team-Member #1 - Govind Sreekar Garimella - 50622437
Team-Member #2 - Deepanathan Rajendiran - 50624197

1. Introduction

This checkpoint report summarizes the current progress of a reinforcement learning (RL) project that focuses on learning insulin dosage strategies from real-world diabetic patient data. The ultimate goal is to train an agent that can recommend appropriate insulin dosage actions—decrease, steady, or increase—based on a patient's glucose levels. We aim to mimic the decision-making behavior of medical professionals using historical data and train a Q-learning agent within a custom OpenAI Gym environment.

The diabetic dataset used comes from clinical records, where each patient entry includes various attributes, including glucose levels, insulin adjustments, and readmission status. For this project, we focused on three key features:

- Maximum glucose serum value (max_glu_serum)
- Insulin action (insulin)
- Patient outcome (readmitted)

From these, we built a simplified environment that captures the relationship between glucose levels and insulin actions, using patient outcomes to guide reward assignment.

2. Data Preprocessing

The dataset was first filtered to include only entries where an insulin action was taken (i.e., insulin was not 'No'). We then mapped the max_glu_serum values to numeric glucose levels using a defined mapping:

- 'None' → NaN (ignored)
- 'Norm' → 100
- '>200' → 250
- '>300' → 350

Only entries with non-null glucose values were retained. The insulin values were mapped to integers to represent actions:

- Down → 0
- Steady → 1
- Up → 2

The reward system was designed based on patient readmission outcomes. Patients readmitted within 30 days (<30) or after 30 days (>30) were considered negative outcomes, and those not readmitted (NO) were treated as positive. Thus, the reward mapping was:

- readmitted == NO → +1
- readmitted == <30 or >30 → -1

This allowed us to create a dataset suitable for training an agent in an RL setting, focusing only on glucose level (state), insulin action (action), and readmission-based reward.

3. Custom Gym Environment

We created a class RealPatientDosageEnv that inherits from gym.Env. This environment:

- Normalizes glucose levels to fall between 0 and 1 (divided by 400).
- Provides a discrete action space with 3 choices (decrease, steady, increase).
- Steps through the dataset sequentially.
- Penalizes incorrect dosage choices slightly by reducing the reward by 0.1.

The environment simulates episodes by selecting a random starting point in the dataset. At each step, it compares the agent's chosen action with the true insulin action from the data and provides a reward based on whether the action matches and the patient outcome.

4. Q-Learning Agent Design

We implemented a tabular Q-learning agent with the following configuration:

- State space: Discretized glucose levels into 20 bins.
- Action space: 3 discrete insulin dosage choices.
- Learning rate (alpha): 0.1
- Discount factor (gamma): 0.9
- Exploration rate (epsilon): 0.1

The agent learns by interacting with the environment and updating its Q-table. The Q-table has a shape of (20, 3), with rows representing discretized glucose bins and columns representing actions.

5. Training the Agent

The agent was trained over 5000 episodes. In each episode, the agent began at a random position in the data and moved through entries one by one, selecting actions, receiving rewards, and updating the Q-table.

After training, the Q-table was converted to a DataFrame for inspection. The Q-values represent the agent's learned preference for each action at each glucose level bin.

6. Results and Analysis

Here is a snapshot of the resulting Q-table (selected values shown for brevity):

Glucose Bin (Discretized)	Action: Decrease	Action: Steady	Action: Increase
0	0.000000	0.000000	0.000000
1	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000
5	0.205712	0.226730	0.148302
6	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000
9	0.000000	0.000000	0.000000
10	0.000000	0.000000	0.000000
11	0.000000	0.000000	0.000000
12	0.093463	-0.019698	0.253301
13	0.000000	0.000000	0.000000
14	0.000000	0.000000	0.000000
15	0.000000	0.000000	0.000000
16	0.000000	0.000000	0.000000
17	-0.301531	-0.167676	0.063904
18	0.000000	0.000000	0.000000

Glucose Bin	Decrease	Steady	Increase
5	0.206	0.227	0.148
12	0.093	-0.020	0.253
17	-0.302	-0.168	0.064

Most other bins (17 out of 20) had zero Q-values for all actions. This indicates that the agent did not receive enough training data for those glucose levels or did not visit them frequently during exploration.

Interpretation:

- **Bin 5**: The agent is slightly more confident in taking a **steady** action here, with close values for decrease and increase.
- **Bin 12**: The Q-values favor **increasing** the dosage, which makes sense if this bin corresponds to high glucose levels.
- **Bin 17**: All actions resulted in negative values, suggesting poor outcomes for patients with those glucose levels, regardless of action. This could be due to an imbalanced or noisy data segment.

7. Observations and Next Steps

- **Sparse Q-table**: Most of the state space remains unexplored or under-trained. This might be due to the small effective sample size per glucose bin. We are using only a subset of real-world data with specific filters, which further reduces variety.
- **Skewed Data**: Some insulin actions are likely over-represented in the dataset. We may need to rebalance or augment the dataset or try quantile binning instead of uniform binning to ensure equal distribution across glucose levels.
- **Limited Learning**: While the agent has learned something for a few glucose bins, the knowledge is limited. Increasing exploration rate (epsilon) temporarily during early episodes could help.
- **Reward Design**: The reward system is simple but effective for this stage. However, we may improve it by assigning graded penalties based on how different the agent's action is from the actual one (e.g., difference of 2 vs. 1 in dosage levels).

8. In Progress

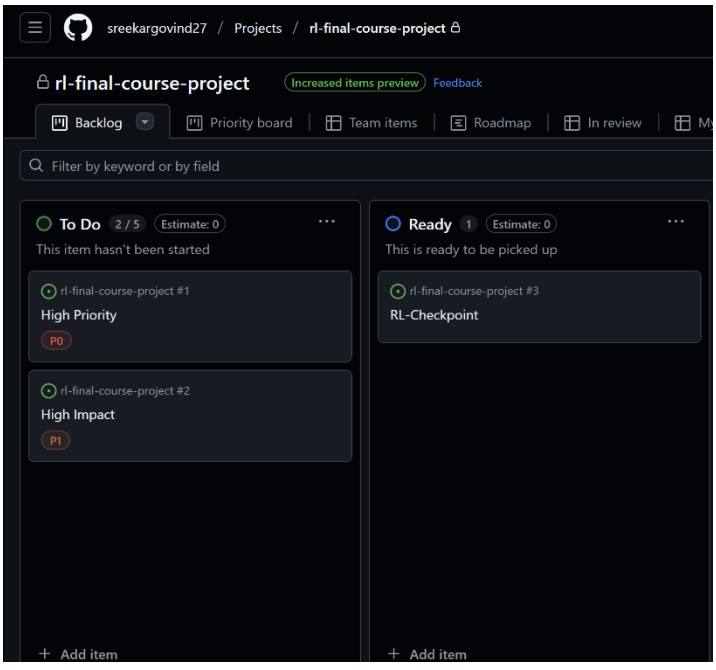
The following items are being worked on:

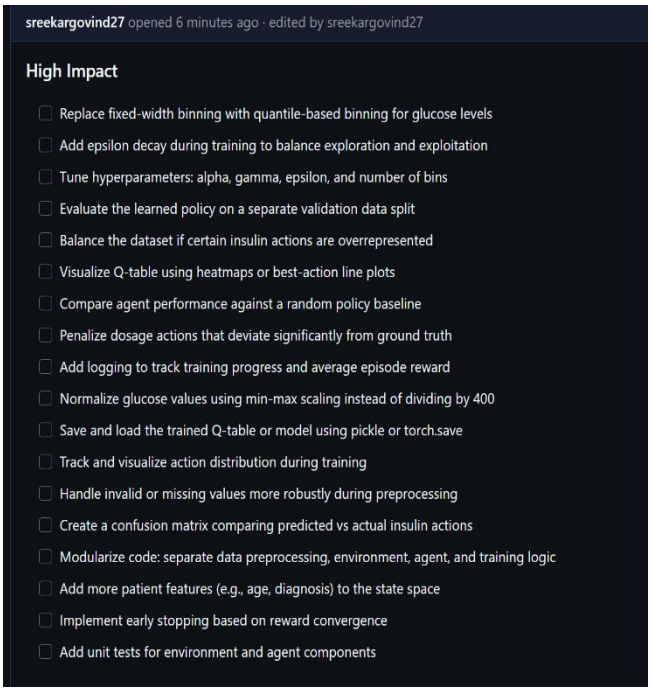
- Plotting best actions across glucose bins for interpretability.
- Evaluating the policy with greedy action selection (ε=0).
- Improving binning strategy using quantiles or Gaussian basis.
- Possibly introducing a DQN to handle continuous glucose levels without binning.

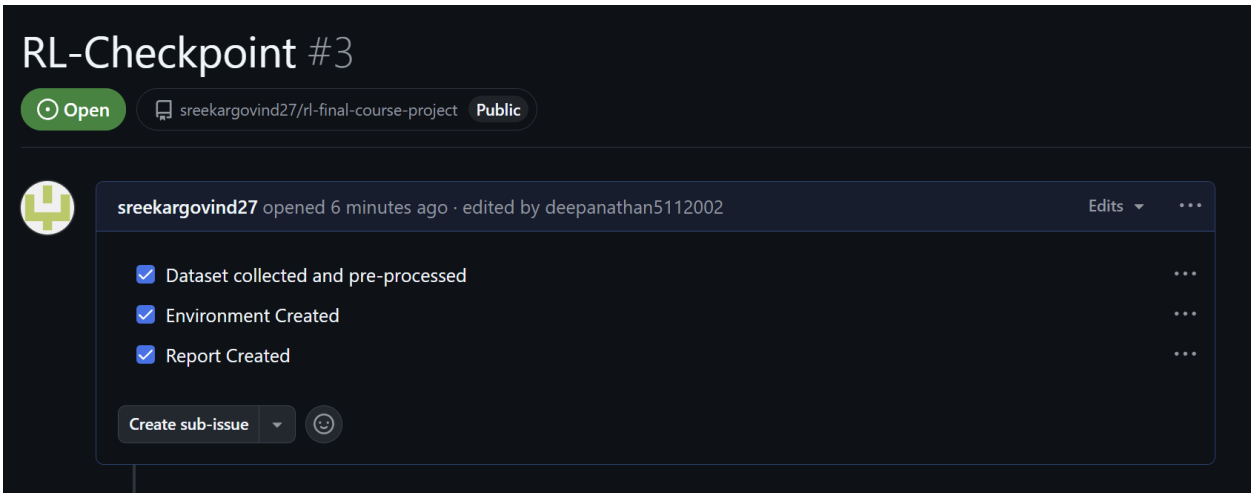
This checkpoint reflects that while the RL pipeline is functional and learns from real patient data, improvements are needed in state coverage, reward shaping, and data balancing to scale the learning further.

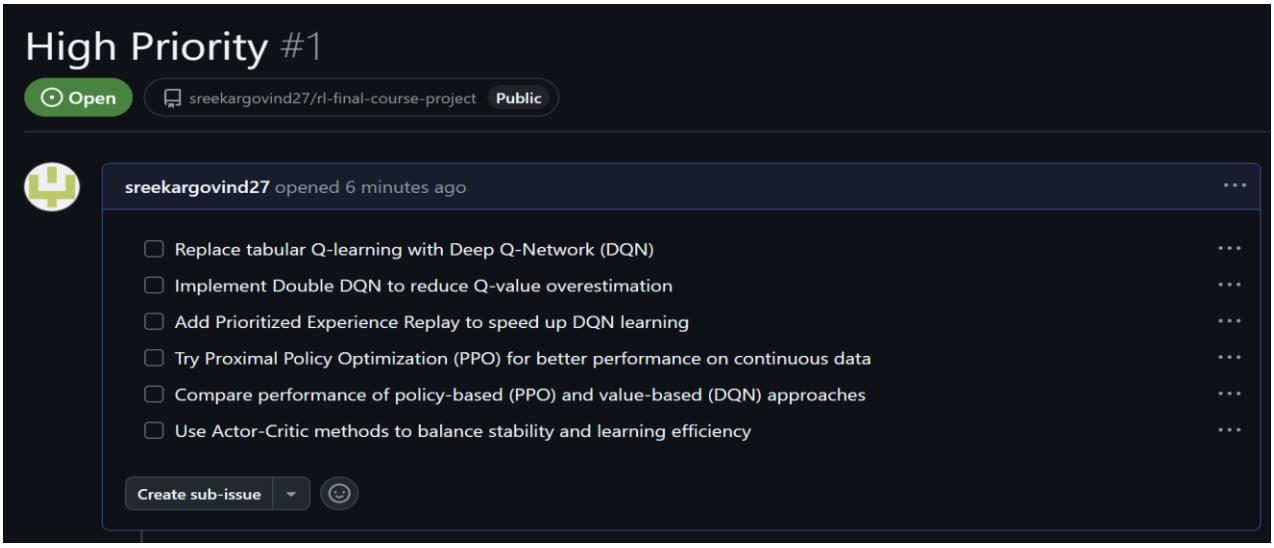
9. Github Updation

This is our github project management dashboard. The Images below show the progress of what we did, and what we will do.






RL-Checkpoint #3
Open
sreekargovind27/rl-final-course-project
Public
sreekargovind27 opened 6 minutes ago · edited by deepanathan5112002
Edits
Dataset collected and pre-processed
Environment Created
Report Created
Create sub-issue


High Priority #1
Open
sreekargovind27/rl-final-course-project
Public
sreekargovind27 opened 6 minutes ago
Replace tabular Q-learning with Deep Q-Network (DQN)
Implement Double DQN to reduce Q-value overestimation
Add Prioritized Experience Replay to speed up DQN learning
Try Proximal Policy Optimization (PPO) for better performance on continuous data
Compare performance of policy-based (PPO) and value-based (DQN) approaches
Use Actor-Critic methods to balance stability and learning efficiency
Create sub-issue