Midterm Exam.

High performance Multiagent Q-learning CUDA program.

Q & A

ECE 277

Cheolhong An

- **Would learning rate schedule be considered time related information?**

If the learning rate (epsilon, alpha or any parameters) is adjusted with agent_adjustespsilon(), It is ok.

but You shouldn't utilize an internal counter or the number of calls to estimate learning time.

If you adaptively change the epsilon or any parameters based on the number of call (episode), then It uses an internal counter. **You should not change the learning rate based on the episode.**

The whole point is to learn without time information.

- An example code in violation (You shouldn’t use the time-based information, In this case, the epsilon is used as a counter, thus, the epsilon includes time-information)

```
if (epsilon > 0.9)
  epsilon -= 0.002;
else if (epsilon > 0.8)
  epsilon -= 0.004;
else if (epsilon > 0.75)
  epsilon -= 0.004;
else
  epsilon -= 0.01;
```

- **Is early stopping okay? That is if we are tracking training accuracy internally, and stop updating the table if accuracy gets worse?**

Update policy is your decision as long as it doesn't utilize time information.

- **Can we use learning rate/epsilon scheduling, which will require tracking of count of the states visited and actions performed? Or is this a violation of using some sort of time information?**

You can not maintain the count during learning. It violates. Basically, you cannot maintain temporal history except the Q-table.

I do not recommend this kind of tweaking approaches instead of focusing on how to efficiently process data

- **I am wondering whether the number of step within each episode belongs to time related information.**

The number of steps within each episode can be used (ex. identify agents how far from an initial position) But you shouldn't accumulate or utilize difference the number of steps between episodes.

- **I came across several variants of Q learning algorithm on several research papers. Can we use them ?**

No

- **Since we will have 4 minutes training time and after that is the real test. I understand that we still need greedy method for training but for the real test, do we need to adjust the code so that our updated Q-table is retained after the training?**

Yes. However, your agent should not know whether it is in training or testing. You shouldn't use any time information.

- **Based on my observation on the code, the qlearning kernel functions are mainly operating on global memory, with few interaction with shared memory or warp parallelism. In such case, should we list down each line of code that we changed compared to lab3 code, no matter the change is using the tuning technique from the lecture?**

No, not line by line. Just describe key changes to improve performance. Even you can list approaches used in lab3.

ex) if you use warp parallelism, then list what you used and where to use and etc (source code line number).

- **Are we still using 0.1 as the minimum value of epsilon for lab 3?**

Epsilon should be 0 to 1 range. (no minimum requirement)

- **I notice in the larger game, when the flag is at a corner of the board, many agents not near the flag are not able to explore a path to the flag before they hit a mine. As a result, a lot of times, my multi-agent learning will result in stucking agents (some agents cluster at certain region surrounded by mines), and my FA rate is barely above 20% even thought sometime it can solve the problem. I am wondering if such a result make sense. Should the correct implementation always results in 80% FA as mentioned in the other thread?**

Your agent doesn't explore enough. That might happen. If It happens, you should explore more instead of too quickly greedy.

- 1. what does the test mode mean, and what does "it finishes the program after all agents become inactive state or 1 minutes." Does it mean that program has 4 minutes to updating the qtable, then in "test mode", environment will use the qtable to check its correctness or performance?

No. From the agent point of view, there is no difference between the training mode and the testing mode. The environment just gathers the flag information during the testing mode. If you run your problem code in the release mode, you will see there is no difference except your program finishes after the testing mode (no new episode even AA lest than 20%). Also, check the exam Q&A for more examples.

- 2. what does "based on the total positive rewards after two runs" mean? Does it mean the environment will count agents that reach the flag in "test mode" within 1 minutes?

Yes. We will run your program two times for the same map: (training + testing) x 2 and your score is the total positive rewards after two runs.

- 3. Some students expressed that their program will converge in a great state less than 4 minutes, even without changing anything from lab3's code. Students asked if it is normal?

I cannot tell exactly whether it is normal or not.

However, the key point is that It is not only convergence but also what methods students can apply from the knowledge of this course. These will be listed in your report. You should list every enhancement and idea in detail. Nothing can harm with more listing.

- “So if several of us finished the training and have all the agents caught the flag (so several of us get 100% accuracy in the test mode), how are we supposed to be ranked (are we given equal ranking or are we differentiated by the time spent to complete one test episode)? “

As I denoted in the exam description, the rank is only based on the total positive rewards. (no time dependence in the test). If there is an equal score, all get the same rank. The rank is only based on total positive rewards (No dependence on the number of students at the same rank) and 0.5 points decrease every rank up to 3 points.

- I'm wondering how can I see the number of flags that the 512 agents are collecting during training and testing time

The environment does not provide the information, but the agent can gather it for only your own purpose.

- Should it be a formal multi-page report with introduction, methods, explanations etc.? Or can I just give a simple bullet point list of changes and explanations?

There is no formal form, but In the exam description, "Write a report about how to enhance your agents
(**Itemize every enhancement in detail**)."

Basically, each item should include a method, explanation, and line number.

Also, you might need to add more information depends on your item.