1. Things seem to start working for both with and without negative rewards after removing the relative-position vector from the input state. (folder – 0)
2. **(Backup-1)** Trained with positive rewards only (rewards clipped from 0 to 2), and reward for discovering new patch added on top of the clipped reward. This seems to learn to collect berries, and does collect about 156 berries in eval after episode 160. But after this the policy seems to degrade (learning rate too high? Or side effect of no rewards or side effect of positive rewards only?).  
   Also note that in this run, the exploration subroutine works in the following way:
   1. **Case-1** The agent is inside the patch with berries visible: The subroutine when invoked will take the same action that it would have taken if invoked outside the patch but only for a single step.
   2. **Case-2** The agent is outside the patch OR no berries are visible: The subroutine does its ***random exploration until it goes inside a patch with berries visible***.

What I noted about this agent that it would not even exploit the current patch. There was too much probability of the agent wondering near the edge of the patch to jump into the exploration subroutine.

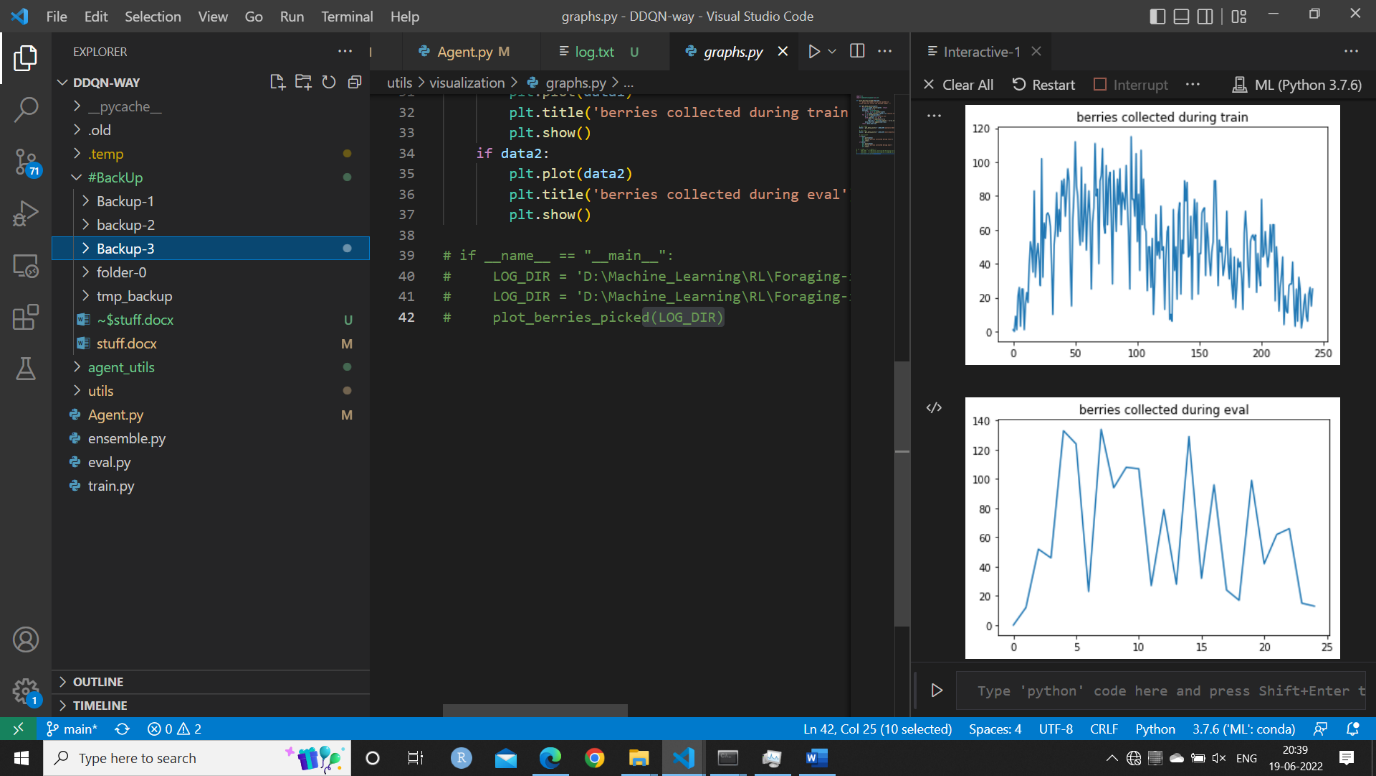
1. **(Backup-2)** Again trained with positive rewards only (clipped in range 0 to 2) with reward for discovering new patch added on top. This again learned to collect berries. The policy of the evaluation fluctuated. ***But it learned to exploit the patches nicely.*** The reason that it did so is I think is the modified exploration subroutine.

The exploration subroutine is as:

* 1. **Continues** the random exploration only until no berries are visible.
  2. **So, the agent cannot activate the exploration for anywhere berries are visible.**
  3. Also, when the random exploration stops – it will almost always stop short of the patch border because berries will be visible even before entering the patch. Agent will need to need to walk into the patch (10s of actions!). Thus, the reward for subroutine is delayed.

I noticed that in evaluation (after many episodes of training) the agents do not want to even use the exploration subroutine!

1. **(Backup-3)**
   1. **(folder-1)** The policy degraded:

The exploration strategy was to allow random exploration until at least one berry is in view, then use the online model to infer which direction to move.

As we can see on the right, the agent quickly learns to collect berries and explore patches, but the policy favors too much of exploration. Also, eventually the policy degraded.

1. **(Backup-4) Multi-resolution Time memories:**

**the berry-field is divided into grids of different resolutions, and we keep a memory of how much time has passed since the agent visited the current cell. (i.e., the agent is only able to access the memory of the current cell)**

* 1. **(Folder 2)** Here we have made an attempt to make the agent avoid getting stuck by using multiple resolution time-memory. So at every making of the state, we would append at random the time-memory from one of the resolutions.  
     The agent did not seem to perform any better.
  2. **(Folder 3)** The entire array of multi-resolution time memories were appended to the state. This was a significant improvement. But the policy collapsed multiple times and though the policy tries to recover by it collapses again and again.
  3. **(Folder 4)** The same agent as in folder-3 but with a much smaller learning rate
  4. **(Folder 5)** This agent is in endowed with a much bigger range of time resolution [20, 50, 100, 200, 400]. This makes the agent go about it's business much smoothly.