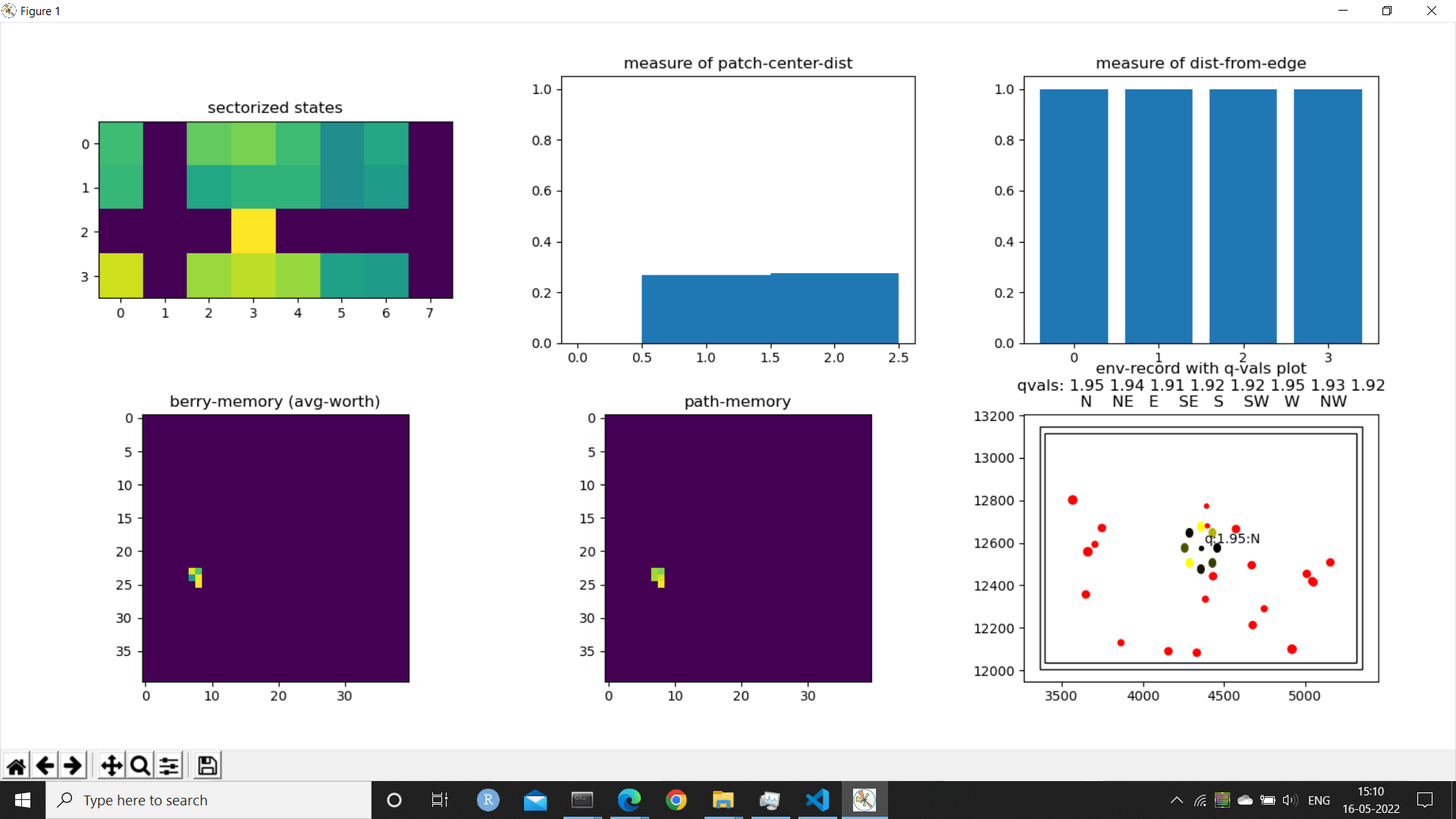
Added memories in form of matrices & scalar – each cell represented a portion of the berryField.

1. Berry-memory matrix: each cell contained the average worth of the observation when the agent was in the corresponding portion, the cell is updated using exponential averaging, as the agent passes through the portion, the cell’s value becomes a somewhat better estimate.

Also, each a cell gets updated whenever the agent gets into the corresponding portion of field.

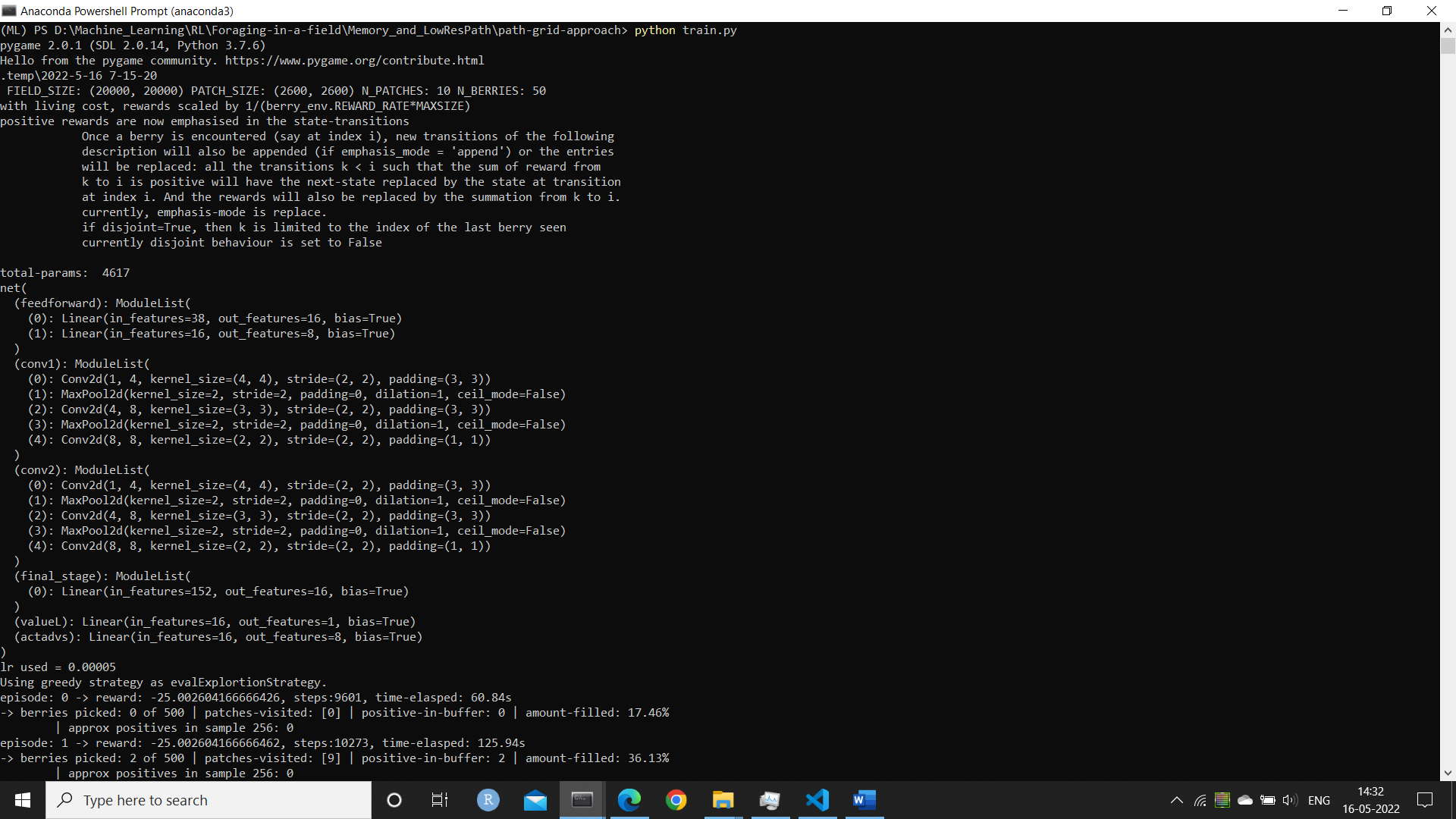
1. Path-memory-matrix: again, each cell represents a portion of the field. For every step (i.e., the end of skip\_steps when makeState is called) the entire memory is decayed by a multiplier. A cell’s value is made 1.0 if the agent is currently in the corresponding portion.
2. Time-memory: This is a scalar, but uses a matrix to store the data. The corresponding cell of the portion of the field the agent is currently in is incremented by a delta. Also, for every step (i.e., the end of skip\_steps when makeState is called) the entire memory is decayed by a factor of 1-delta.



Time-memory

Things tried but have-not yet worked:

1. Flattening the entire state and feeding into a fc network: Too many learnable parameters.
2. (Folder named ‘2’) A fully-connected-feedforward for the sectorized-states + edge-distances + patch-relative + time-memory. Separate conv-layers for each of berry-memory and patch memory. All were concatenated and fed into a dueling-net type of network. The conv-layer outputs were of shape (8,3,3). Total parameters are 4617.



My suspicions:

Perhaps the conv-layers were separated for far too long that the information from the different parts of the states could not be efficiently combined.