Assignment 11

Lilian Kourti
CME 241

Problem 1

See RL-book/_lkourti/Assign11/prob11_1.py for the implementation Tabular MC Prediction. The learning rate was reduced appropriately as a function of number of updates.

Problem 2

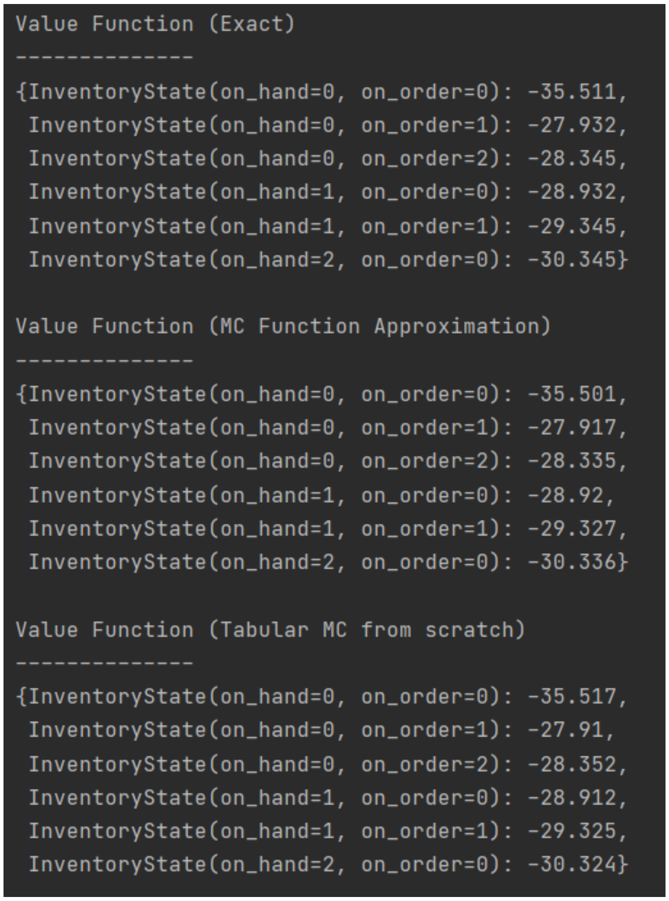
See RL-book/_lkourti/Assign11/prob11_2.py for the implementation Tabular TD Prediction. The learning rate was reduced appropriately as a function of number of updates, as follows:

$$\alpha_n = \frac{\alpha}{1 + (\frac{n-1}{H})}$$

where α_n is the learning rate to be used at the n-th Value Function update for a given state, α is the initial learning rate, H (we call it "half life") is the number of updates for the learning rate to decrease to half the initial learning rate (if β is 1), and β is the exponent controlling the curvature of the decrease in the learning rate.

Problem 3

For the case of the Simple Inventory MRP, we compare the Value Function as derived using: (i) the exact calculation, (ii) the function approximation with Monte Carlo using 10000 traces, and (iii) the tabular Monte Carlo implementation from scratch, again using 10000 traces. See also RL-book/_lkourti/Assign11/prob11_3a.py. Indeed, the three approaches yield almost identical results:

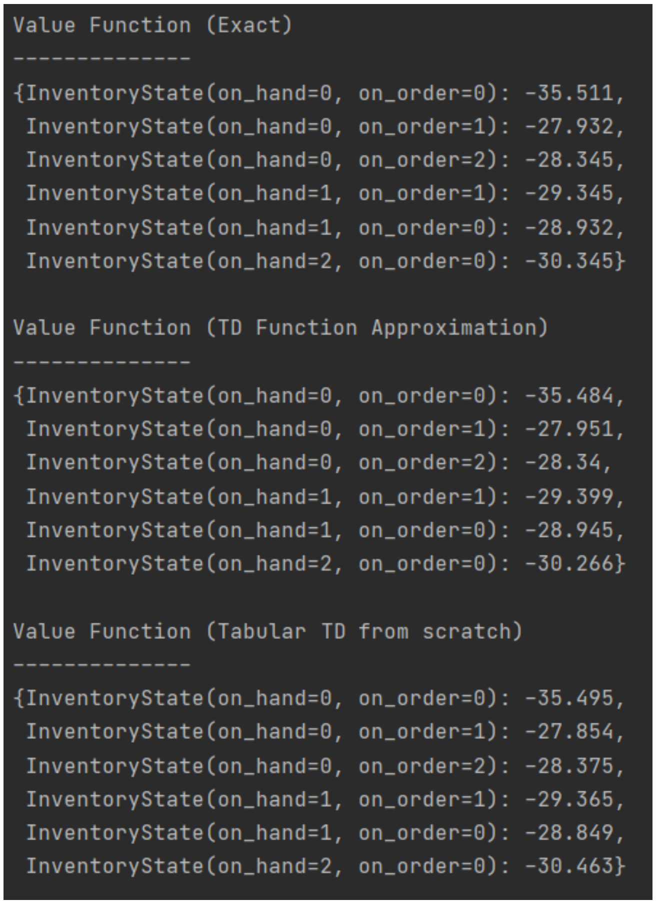


```
Value Function (Exact)
--------------
{InventoryState(on_hand=0, on_order=0): -35.511,
 InventoryState(on_hand=0, on_order=1): -27.932,
 InventoryState(on_hand=0, on_order=2): -28.345,
 InventoryState(on_hand=1, on_order=0): -28.932,
 InventoryState(on_hand=1, on_order=1): -29.345,
 InventoryState(on_hand=2, on_order=0): -30.345}

Value Function (MC Function Approximation)
--------------
{InventoryState(on_hand=0, on_order=0): -35.501,
 InventoryState(on_hand=0, on_order=1): -27.917,
 InventoryState(on_hand=0, on_order=2): -28.335,
 InventoryState(on_hand=1, on_order=0): -28.92,
 InventoryState(on_hand=1, on_order=1): -29.327,
 InventoryState(on_hand=2, on_order=0): -30.336}

Value Function (Tabular MC from scratch)
--------------
{InventoryState(on_hand=0, on_order=0): -35.517,
 InventoryState(on_hand=0, on_order=1): -27.91,
 InventoryState(on_hand=0, on_order=2): -28.352,
 InventoryState(on_hand=1, on_order=0): -28.912,
 InventoryState(on_hand=1, on_order=1): -29.325,
 InventoryState(on_hand=2, on_order=0): -30.324}
```

For the case of the Simple Inventory MRP, we compare the Value Function as derived using: (i) the exact calculation, (ii) the function approximation with TD using 100000 traces, and (iii) the tabular TD implementation from scratch, again using 100000 traces. See also RL-book/_lkourti/Assign11/prob11_3b.py. Indeed, the three approaches yield almost identical results:



```
Value Function (Exact)
--------------
{InventoryState(on_hand=0, on_order=0): -35.511,
 InventoryState(on_hand=0, on_order=1): -27.932,
 InventoryState(on_hand=0, on_order=2): -28.345,
 InventoryState(on_hand=1, on_order=1): -29.345,
 InventoryState(on_hand=1, on_order=0): -28.932,
 InventoryState(on_hand=2, on_order=0): -30.345}

Value Function (TD Function Approximation)
--------------
{InventoryState(on_hand=0, on_order=0): -35.484,
 InventoryState(on_hand=0, on_order=1): -27.951,
 InventoryState(on_hand=0, on_order=2): -28.34,
 InventoryState(on_hand=1, on_order=1): -29.399,
 InventoryState(on_hand=1, on_order=0): -28.945,
 InventoryState(on_hand=2, on_order=0): -30.266}

Value Function (Tabular TD from scratch)
--------------
{InventoryState(on_hand=0, on_order=0): -35.495,
 InventoryState(on_hand=0, on_order=1): -27.854,
 InventoryState(on_hand=0, on_order=2): -28.375,
 InventoryState(on_hand=1, on_order=1): -29.365,
 InventoryState(on_hand=1, on_order=0): -28.849,
 InventoryState(on_hand=2, on_order=0): -30.463}
```