Day: RL Notes Date:

AI

to mimic the behavior of intelligent human beings
(simulation of human intelligence in machines)

Rule based (Deduction	Example based (Induction)
rules or logic defined by humans	Machine learning to learn patterns and behaviors from data or examples Subcategories - Supervised Learning - Unsupervised learning - Reinforcement learning

In reinforcement learning, reinforcement refers to the feedback (rewards or penalties) that an agent receives as it interacts with its ~~enf~~ environment, guiding it to improve its decisions or behavior.
(intelligent)

An agent in AI is an autonomous entity that perceives its environment, makes decisions based on that perception, and takes actions to achieve specific goals or objectives.

Day:_____ Date:_____

Goal:

Reward:

A reward is a numerical signal that provides feedback to an agent about the success of its actions in achieving a desired goal.

Goal:

It is maximizing the total reward (cumulative reward) over time.

Rewards provide (immediate feedback) about whether the actions are beneficial (moving it closer to the goal) or detrimental (moving it away from the goal)

policy: optimal behavior
strategy that defines how an agent acts in an environment

Difference between supervised and reinforcement

Train on a labeled dataset	learn to play by trial and error
reducing errors	maximizing its rewards
	delayed feedback

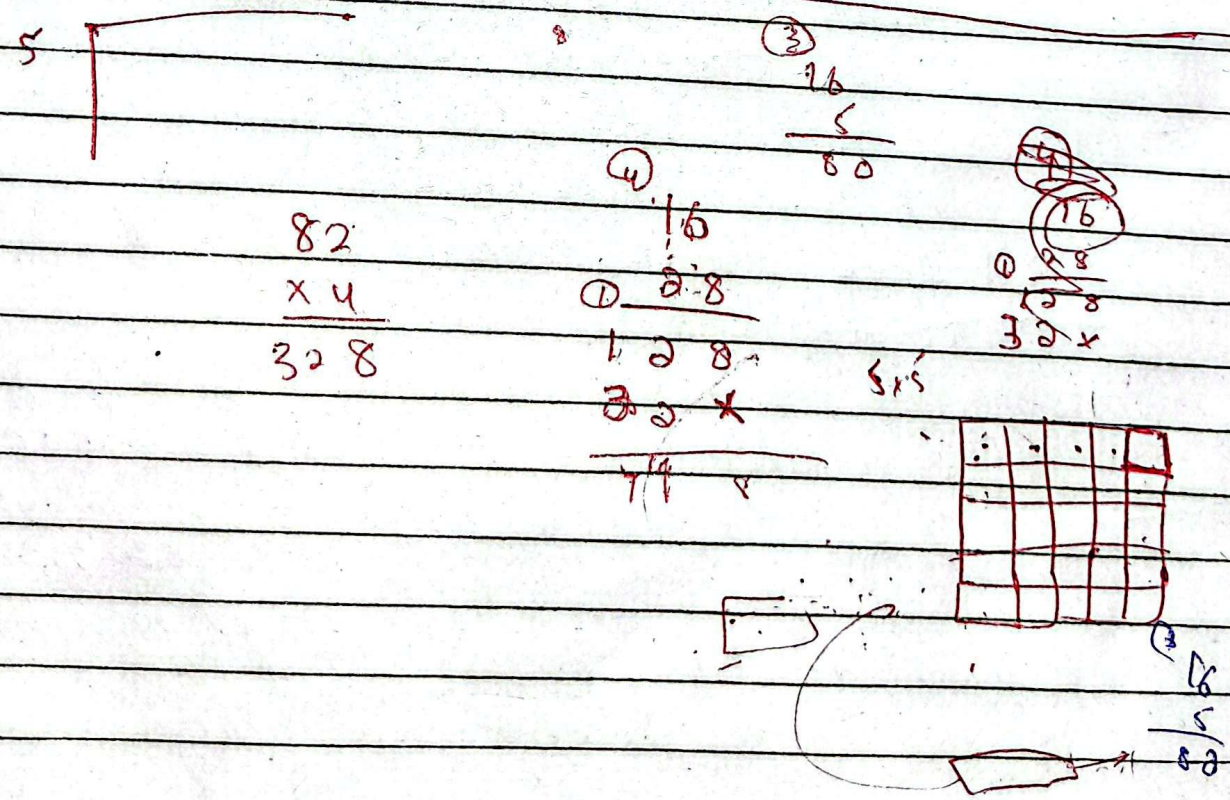
Date:

Sequential Decision Making

- Goal: select actions to maximize total future reward
- Actions may have long term consequences
- Rewards may be delayed

Agent at each step t
- Execute action A_t
- Receive observation O_t
- Received scalar reward R_t

Environment,
- Receive action A_t
- Emits observation O_{t+1}
- Emits scalar reward R_{t+1}



Day: Date:

States:

5x5 grid - 25 states
5 passenger states (4 loc + inside taxi)
4 destination pairs
25 * 5 * 4 = 500

(Taxi (3,1))

State 328

~~(5 * row~~
~~row = no~~
((row * 5) + column) *
(5 + pass_locn) * 4 +
destination

Taxi at (3,1) P: Y D: R

Pass loc: (R:0, G:1, Y:2, B:3, In taxi: 4)

Destloc: (R:0, G:1, Y:2, B:3)

(3 * 5 + 1) * (5 + 2) * 4 + (0)
16 * (7 * 4)
16 * 28

State No = ((Taxi row * 5 + taxi_column) x 5
+ passenger_index) x 4 + Destination

= (((3x5+1) x 5 + 2) * 4) + 0
= (16 x 5 + 2) x 4)
= (82 x 4)
= 328

4 | 328 (82)
28
0

Destination: 0

5 | 82 (16)
80
2

Passenger loc: 2

5 | 16 (3)
15
1

(3,1)

Date:

policy defines how the agent selects action in each state to maximize cumulative rewards

Solution Methods in RL

- tabular solution
  - MAB
  - MDP
- Approximate solution

Multi Armed bandits

- The simplest RL problem
- One state
- Actions don't have delayed effects

The term "bandit" comes from the analogy of a slot machine, also known as a one-armed bandit, because it has a lever (arm) and "steals" money from players over time

True value of an action:
is the expected value of the rewards the agent can get for that action.

$$q_*(a) = \mathbb{E}[R_t | A_t = a], \quad \forall a \in \{1, \dots, k\}$$

The multi-armed bandit problem involves choosing between multiple actions (like slot machine arms) to maximize reward over time. It's named after slot machines ("one-armed bandits") imagine choosing which of multi-b slot machines to play.

Day: Date:

Estimated value

$$Q(a_1) \approx q^*(a_1)$$

Action value methods

The action value method estimates the expected reward of each action based on past experiences

types

Sample Average Method

Incremental average method

$$Q_n = \frac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$$

$$Q_{n+1} = Q_n + \frac{1}{n}[R_n - Q_n]$$

$$Q_{n+1} = Q_n + \frac{1}{n}(R_n - Q_n)$$

New Estimate = Old Estimate + Step size * (Target - OldEstimate)

Exploration vs exploitation dilemma:

It involves deciding when to explore new actions to gather more information about their potential rewards (exploration) versus when to choose the action that currently has the highest estimated value (exploitation)

$\varepsilon \rightarrow$ exploration ε value between

$1-\varepsilon \rightarrow$ exploitation. 0 and 1

We choose ~~act~~ random action with certain probability ε and greedy action on $1-\varepsilon$

Day:_____ Date:_____

10 armed Bandit Problem

Stationary Non Stationary

$$Q_{n+1} = Q_n + \frac{1}{n}[R_n - Q_n]$$

$$Q_{n+1} = Q_n + \alpha(R_n - Q_n)$$

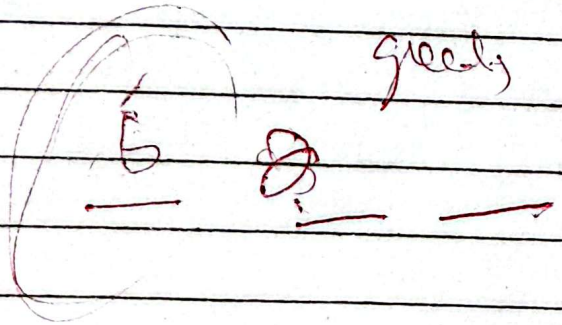
k = 10

10 armed testbed

In the 10 armed test bed the true values $q^*(a)$ for each of the ten actions are drawn from a normal distribution with mean 0 and variance 1.

For the epsilon-greedy algorithm, different epsilon values control the probability of exploring random actions.

An initial optimistic value can be assigned to encourage exploration



Day:_____ Date:_____

for k = 10 (10 actions)
and for the actions 1000
no.

for each action

Create 10 actions
For each action we max run (and get reward value 1000 times)
We repeat the whole episode 2000 time

10 arms
get rewards true value from NoD

Run Exp
Initialize
how many times to repeat (n = 2000)
how many steps per run (e = 1000)
Create array to store mean rewards

For each run (200 times):
create 10 actions with random true values
For each time step (1000 times):
choose action
If random < epsilon:
select random
else
select action with highest estimate val
Get reward:
from normal distribution
Update estimate:
new estimate = old estimate + (reward - old estimate) / time selected

Day:_____ Date:_____

$$Q_{n+1} = \frac{1}{n-1} \sum_{i=1}^{n-1} R_i$$

$$Q_{n+1} = \frac{1}{n} \sum_{i=1}^{n} R_i$$

$$Q_{n+1} = \frac{1}{n} \left[\sum_{i=1}^{n-1} R_i + R_n \right]$$

$$Q_{n+1} = \frac{1}{n} \left[\frac{n-1}{n-1} \sum_{i=1}^{n-1} R_i + R_n \right]$$

$$= \frac{1}{n} \left[(n-1) Q_n + R_n \right]$$

$$= \frac{1}{n} \left[n Q_n - Q_n + R_n \right]$$

Incremental

$$Q_{n+1} = Q_n + \frac{1}{n} \left[R_n - Q_n \right]$$

Steps	Action taken	Reward	Sample Avg	Incremental
1	A1	2	2	2
2	A2	5	5	5
3	A1	4	3	3
4	A3	1	1	1
5	A1	3	3	3
6	A2	6	5.5	5.5
7	A3	4	2.5	2.5
8	A2	3	4.67	4.67

Day:_____ Date:_____

$$Q_{n+1} = Q_n + \alpha [R_n - Q_n]$$

$$= Q_n + \alpha R_n - \alpha Q_n$$

$$Q_{n+1} = (1-\alpha) Q_n + \alpha R_n$$

$$Q_{n+1} = \alpha R_n + (1-\alpha)(Q_n)$$

$$= \alpha R_n + (1-\alpha)(\alpha R_{n-1} + (1-\alpha) Q_{n-1})$$

$$= \alpha R_n + \alpha(1-\alpha) R_{n-1} + (1-\alpha)^2 Q_{n-1}$$

$$= \alpha R_n + \alpha(1-\alpha) R_{n-1} + \alpha(1-\alpha)^2 Q_{n-2} + (1-\alpha)^3 Q_{n-3}$$

$$= \alpha R_n + \alpha(1-\alpha) R_{n-1}) + \ldots$$

$$\therefore (1-\alpha)^n Q_1$$

$$= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha (1-\alpha)^{n-i} R_i$$

Date:

- Optimistic Initial Values

We set initial values other than 0

This optimism encourages action-value methods to explore.

UCB

Upper Confidence Bound Action Selection

select actions according to

$$A_t = \underset{a}{\operatorname{argmax}} \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right]$$

c controls the degree of exploration.

if $N_t(a) = 0$ then a is considered to be a maximizing action.

$N_t(a)$ denotes the no of times that action a has been selected prior to time t.

each time an action is selected its uncertainty reduced.

Day:_____ Date:_____

Optimistic Initial values Involve setting the initial estimates of action values to big high values instead of 0.

Encourages exploration

helps balance exploration and exploitation without requiring an explicit exploration parameter.

UCB

$$A_t = \text{argmax}_a \left(Q(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right)$$

$Q(a)$: Current estimate of the action value.

c: Exploration parameter

Int: Natural log of the total number of steps taken so far

$N_t(a)$: Number of times action a has been chosen up to time(t)

Actions that are less explored (small $N_t(a)$) have a higher uncertainty term, encouraging the agent to try them.

ay:_____ Date:_____

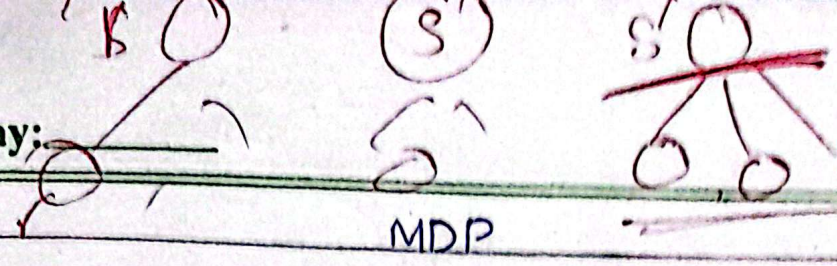
MAB	MDB
1) Only one state, no state transition.	Involves multiple states; the state changes based on actions.
~~Action influences future.~~	
2) Actions provide immediate rewards only, no effect on future rewards.	Action influences future states and long term rewards
3) No delayed effects, rewards are immediate.	Actions may have delayed effects.
4) Maximize immediate total rewards from repeated actions.	Maximize cumulative rewards over a sequence of steps.
5) Simple problem, no state transitions or policies	More complex due to state transitions and policies
6) Focuses on the explorain and exploitation trade off to find the best action	Requires learning an optimal policy for selecting actions in each state,

$P(s'|s,a)$:

$V(s,a)$:

$$\sum_r \sum_{s'} p(s',r|s,a) \sum_r p(s',r|s,a)$$

Day: Date:



MDP

$$p(s'/s,a) = \sum_r p(s',r/s,a)$$

$$r(s,a) = \sum_r r \cdot \sum_{s'} p(s',r/s,a)$$

Expected reward

$$r(s,a,s') = \sum_r r \frac{p(s',r/s,a)}{p(s'/s,a)}$$

$$p(s'/s,a) = \sum_r p(s',r/s,a)$$

$$r(s,a) = \sum_r r \sum_{s'} p(s',r/s,a)$$

$$r(s,a,s') = \sum_r r \frac{p(s',r/s,a)}{p(s'/s,a)}$$

$$r(s,a,s') = \sum_r r \frac{p(s',r/s,a)}{p(s'/s,a)}$$

Day:_____ Date:_____

$S = \{s_0, s_1, s_2\}$

$A = \{a, b\}$

s_0,

a

$p(s_1, r=2 \mid s_0, a) = 0.4$

$p(s_0, r=0 \mid s_0, a) = 0.6$

b

$p(s_1, r=+1 \mid s_0, b) = 0.5$

$p(s_2, r=3 \mid s_0, b) = 0.5$

s_1

a:

$p(s_0, r=+1 \mid s_1, a) = 1$

b:

$p(s_2, r=2 \mid s_1, b) = 1.0$

s_2 zero

$$p(s' \mid s, a) = \sum_r p(s', r \mid s, a)$$

$$p(s_0 \mid s_1, a_1) = \sum_r p(s', r \mid s, a) = 1$$

$p(s_1 \mid s_0, a$

Day:_____ Date:_____

S_0: E

S_1: P

S_2: F

a: Load

b: Drive

$S_0 \rightarrow a$

~~$S_1 \mid S_0, a$~~

$(S_1, r=2 \mid S_0, a) = 0.4$

$(S_2, r=3 \mid S_0, a) = 0.3$

$(S_0, r=0 \mid S_0, a) = 0.3$

$(S_1, r=1 \mid S_0, b) = 0.5$

$(S_2, r=5 \mid S_0, b) = 0.2$

$(S_0, r=0 \mid S_0, b) = 0.3$

$(S_2, r=4 \mid S_1, a) = 0.6$

$(S_0, r=1 \mid S_1, a) = 0.3$

$(S_1, r=0 \mid S_1, a) = 0.1$

$(S_0, r=2 \mid S_1, b) = 0.2$

$(S_2, r=3 \mid S_1, b) = 0.7$

$(S_1, r=0 \mid S_1, b) = 0.1$

Day: Date:

~~set~~ S = {high, low}

A = {search, wait, recharge}

A(s) : A(high) {search, wait}

A(low) {search, wait, recharge}

rescued → low reward

high → ~~sea~~ wait & low → α

~~P(s'|s,a) = Σ_r P(s',r|s,a)~~

$$p(s'|s,a) = \sum_r p(s',r|s,a)$$

~~pt~~

$$r(s,a) = \sum_r \sum_{s'} p(s',r|s,a)$$

$$r(s,a,s') = \sum_r r\, p(s',r|s,a)$$

~~st~~ s	a	s'	$\frac{p(s',r\|s,a)}{p(s'\|s,a)}$	r(s,a,s')
high	search	high	α	r_{search}
high	search	low	1-α	r_{search}
~~high~~ low	~~sea~~ search	low	~~β~~ β	r_{search}
low	search	high	1-β	-3
high	wait	high	1	r_{wait}
high	wait	low	0	-
low	wait	high	0	
low	wait	low	1	-
low	~~recharge~~	high	1	r_{wait}
low	recharge	low	0	0

Day:_____ Date:_____

Lecture 7

A_1 A_2 A_3

explo

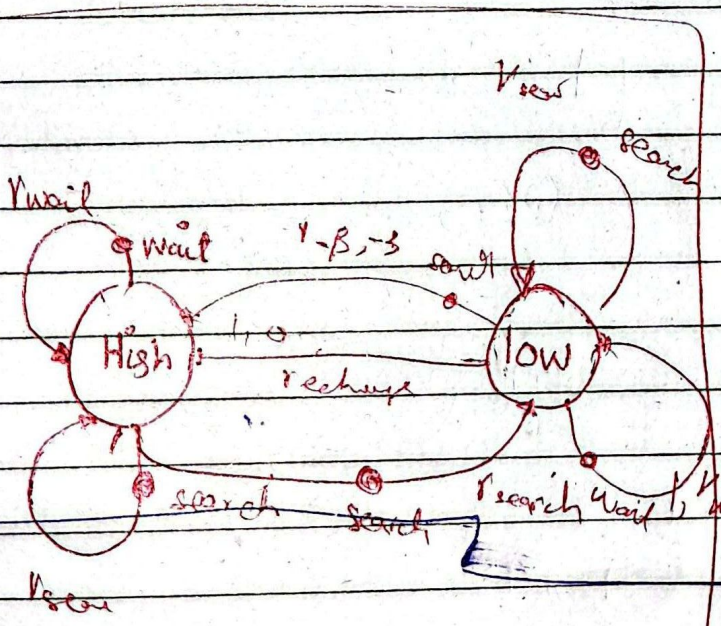
explor

0.8

Optimistic initial values

Setting the Initial values other than 0 can encourage exploration

A_1	A_2	A_3
5	5	5
t=1	5	2
t=2		



$$Q_{t+1} = Q_t + \frac{1}{n}[R_t - Q_t]$$

$$= 5 + \frac{1}{2}[2-5]$$

$$= 5 - 1.5$$

Day:_____ Date:_____

RL rewards

$$p(s'|s,a) = \sum_r p(s',r|s,a)$$

$\sum_r p(s_0,r|s_1,a) = 0.3$ $\quad r(s,a) = \sum_r r \cdot \sum_{s'} p(s'|s,a)$

$p(s_0,r|s_1,b) = 0.2$

a	$p(s_0\|s_1,a) = 0.3$	$p(s_1\|s_0,a) = 0.4$
	$p(s_0\|s_1,b) = 0.2$	$p(s_1\|s_0,b) = 0.5$
a	$p(s_0\|s_0,a) = 0.3$	$p(s_1\|s_1,a) = 0.1$
	$p(s_0\|s_0,b) = 0.3$	$p(s_1\|s_1,b) = 0.1$
a	$p(s_0\|s_2,a) = 0$	$p(s_1\|s_2,a) = 0$
	$p(s_0\|s_2,b) = 0$	$p(s_1\|s_2,b) = 0$

$p(s_2\|s_0,a) = 0.3$	~~r(s,a)~~
$p(s_2\|s_0,b) = 0.2$	
$p(s_2\|s_1,a) = 0.6$	
$p(s_2\|s_1,b) = 0.7$	
$p(s_2\|s_2,a) = 0$	
$p(s_2\|s_2,b) = 0$	

Day:_____ Date:_____

Episodic: $G_t = \sum_{t'=t+1}^{T} R_{t'}$

Continuous

$$G_t = \sum_{t'=t+1}^{\infty} R_{t'}$$

$$G_t = \sum_{k=t+1}^{+\infty} \gamma^{\cdot}$$

$$\sum_{t'=t+1} R_{t'}$$

$$G_t = \sum_{k=0}^{+\infty} \gamma^k R_{t+1+k} \qquad 0 < \gamma < 1$$

$$\sum_{t'=t+1}^{\infty} R_{t'}$$

$$G_t = R_{t+1} + \gamma G_{t+1}$$

$$G_t = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$$

$$\sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$$

Undefine if $T = \infty$ and $\gamma = 1$

$\pi(a|s)$

V_π $\quad \pi(a)$

$$V_\pi(s) = \mathbb{E}_\pi(G_t | S_t = s)$$

$$= \mathbb{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right)$$

$$= \pi(a|s) \sum_r \sum_{s'} p(s', r | s, a) \cdot [r + \gamma$$

Day:_____ Date:_____

Policies

$\pi(a|s)$

Value function

$$V_\pi(s) = E_\pi[G_t | S_t = s]$$

$$V_\pi(s) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right]$$

Action value func

$$q_\pi(s,a) = E_\pi\left[\sum_{k=0} \gamma^k R_{t+k+1} | S_t = s, A_t = a\right]$$

$$V_\pi(s) = E_\pi[G_t | S_t = s]$$

$$= E_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s]$$

$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r | s, a)[r + \gamma E_\pi[G_{t+1} | S_{t+1} = s']]$$

$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r | s, a)[r + \gamma V_\pi(s')]$$

Day:_____ Date:_____

Deterministic Policy	Stochastic Policy
A policy map each state to a given action $\pi(s) = a$	A policy assign probabilities to each action in each state $\pi(a\|s)$ $\sum_{a \in A(s)} \pi(a\|s) = 1$

Policies can be only depend on current state

$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$ state value is basically a future reward than an agent expect to receive starting from a particular state

~~State~~ value function $V(s) = E[G_t | S_t = s]$

action value fun $q(\pi, a) = E_\pi[G_t | S_{t+1} = s, A_t = a]$

$$V(s) = \pi(a|s) \sum_r \sum_{s'} p(s', r | s, a) [r + \gamma V(s')]$$

Day:_____ Date:_____

$\gamma = 0.9$

State-value function represent the expected return from a given state under a specific policy.

Action-value function represent the expected return from a given state after taking a specific action, later following a specific policy.

Bellman Equation

$$V_\pi(s) = \mathbb{E}_\pi [G_t | S_t = s]$$

$$q_\pi(s,a) = \mathbb{E}_\pi (G_t | S_t = s, A_t = a)$$

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

$$V_\pi(s) = \mathbb{E}_\pi (R_{t+1} + \gamma G_{t+1} | S_t = s)$$

$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s',r|s,a) [r + \gamma \mathbb{E}_\pi (G_{t+1} | S_{t+1} = s'))$$

$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s',r|s,a) [r + \gamma V_\pi(s')]$$

- $\sum_s \sum_r p(s',r|s,a) = \sum_r \sum_{s'} p(s',r|s,a)$

$$q_\pi(s,a) = \gamma + \gamma \sum_{a'} \pi(a'|s') q_\pi(s',a')$$

$$\sum_r \sum_{s'} p(s',r|s,a)$$

$$[G_t | S_t = s, A_t = a]$$

$$= \sum_r \sum_{s'} p(s',r|s,a)$$

$$[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a]$$

$$= \sum_r \sum_{s'} p(s',r|s,a)$$

$$[r + \gamma \mathbb{E}_\pi (G_{t+1} | S_{t+1} = s')]$$

$$= \sum_r \sum_{s'} p(s',r,s,a)$$

$$[r + \gamma \sum_{a'} \pi(a'|s') \mathbb{E}_\pi [G_{t+1} | S_{t+1} = s', A_{t+1} = a']]$$

Day:_____ Date:_____

In ε-greedy action selection, for the case of two actions and $\varepsilon = 0.5$, what is the probability that the greedy action is selected

$\varepsilon = 0.5$

greedy action $= 1 - \varepsilon + \frac{\varepsilon}{\text{no of actions}}$

$= 1 - 0.5 + \frac{0.5}{2}$

$= 1 - 0.5 + 0.25$

$= 0.5 + 0.25$

$= 0.75$

Day:_____ Date:_____

k : 4 actions

Action → 1, 2, 3, 4

$Q_1(a) = 0$

timestep	1	2	3	4	
A1	1				Q(1) = 1
A2		1			Q(2) = 1
A3		2			Q(2) = 1.5
A4		2			
A5			0		

	1	2	3	4
~~random~~	0	0	0	0
random	1			
random		1		
random: greedy		2		
greedy		2		
random			0	

Day:_____ Date:_____

Flappy bird Game

States
- bird's vertical position (high, medium, low)
- bird's velocity (rising, falling)
- Distance to the next pipe

Actions
Flap (move up)
Do nothing (let the bird fall)

Goal
pass through many pipes as possible without crashing
epsiodes end when bird crashes into the pipe or ground

Rewards
+10 for successfully passing through the pipe
0 for staying alive with passing a pipe
-5 for crashing into a pipe

Space rover
Stats
- grid coordinates
destination
starting point
- hazards
collecting sample

Actions
Move (north, east, west, south
- Collect sample
- Recharge
- Avoid hazards

Rewards:
+15 for collecting sample.
-10 for encoutering hazard
-5 low energy without recharging
0 for exploring when needed sample

Day:_____ Date:_____

Car Refueling Trip

states
- home
- Driving to gas station
  Waiting in line
- Refueling car
- Returning home

- Choose nearest station
- Wait or skip long line
- Refuel
- Return home

Rewards
+10. Refueling successful
-5. Running low on fuel before reaching to station
-3 Wait long in line

s	a	s'	r	p(s', r | s, a)
high	wait	high	r_{wait}	1
high	wait	low	0	-
high	search	high	r_{search}	α
high	search	low	r_{search}	$1-\alpha$
~~high~~	~~recharge~~	~~high~~	~~1~~	
~~high~~	~~recharge~~	~~low~~	~~0~~	
low	wait	high	0	-
low	wait	low	r_{wait}	1
low	search	high	-3	β
low	search	low	r_{search}	$1-\beta$
low	recharge	high	1	0
low	recharge	low	0	-

Day:_____ Date:_____

Car Refueling Trip

states
- home
- Driving to gas station
  Waiting in line
- Refueling car
- Returning home

- Choose nearest station
- Wait or skip long line
- Refuel
- Return home

Rewards
+10 Refuels successfully
-5 Runs low on fuel before reaching the station
-3 Wait long in line

s	a	s'	r	p(s', r | s, a)
high	wait	high	r_{wait}	1
high	wait	low	0	-
high	search	high	r_{search}	α
high	search	low	r_{search}	$1-\alpha$
~~high~~	~~recharge~~	~~high~~	~~1~~	
~~high~~	~~recharge~~	~~low~~	~~0~~	
low	wait	high	0	-
low	wait	low	r_{wait}	1
low	search	high	-3	β
low	search	low	r_{search}	$1-\beta$
low	recharge	high	1	0
low	recharge	low	0	-

Day:_____ Date:_____

$\gamma = 0.5$

$R1 = 1,\ R2 = 2,\ R3 = 6,\ R4 = 3,\ R5 = 2$

$T = 5$

find $G0, G1, \ldots G5$.

$G_t = R_{t+1} + \gamma G_{t+1}$

$G_4 = R_5 + \gamma G_6$

$G_4 = 2 + (0.5)(0) = 2$

$G_3 = R_4 + \gamma G_4$

$= 3 + (0.5)(2) = 4$

$G_2 = R_3 + \gamma G_3$

$= 6 + (0.5)(4) = 8$

$G_1 = R_2 + \gamma G_2$

$= 2 + (0.5)(8) = 6$

$G_0 = R_1 + \gamma G_1$

$= 1 + (0.5)(6) = 4$

$\gamma = 0.9$ $R1 = 2$ followed by an infinite sequence of 7s

Find $G1$ & $G0$

$$G_1 = \sum_{k=0}^{\infty} R_k \cdot \gamma^k$$

$$= R_k \cdot \frac{1}{1-\gamma}$$

$$= 7 \cdot \frac{1}{1-0.9} = 70$$

$G_0 = 2 + (\gamma) 70 = 2 + (0.9)(70) = 65$

Day:_____ Date:_____

$\gamma = 0.8$

$R_1 = 5$

10

$G_0 = \gamma G_1 + R_{t+1}$

$G_1 = R_t \frac{1}{1-\gamma}$

$= \frac{1}{0.2}$

50

5

Cost $50 + 10$

$=$