Introduction

The is a project aimed at creating an autonomous Pacman game where Pacman's movements and decision-making are generated using Reinforcement Learning (MDP and Q-Learning) and Machine Learning. This report provides an overview of the approaches used for the tasks.

Part 1 – Reinforcement Learning (MDP and Q-Learning)

Task 1b (MDP-value iteration)

Approach used

First, to determine the iteration value, we will begin at 100 and increment it by 100 with each iteration (while using the default discount factor, 0.6). We will run the process at least five times (i.e., when the iteration value reaches 500). Subsequently, we will assess whether it has the highest average score compared to the others. If it does, we will continue running it until it converges (i.e., when the average score remains the same or decreases) and select the iteration value with the highest average score. Otherwise, we will stop and select the iteration value with the highest average score.

Once we have determined the iteration value, we will use it to calculate our discount factor by testing each factor between 0.3 and 1 with a step size of 0.05. The discount factor with the highest average score will be chosen.

bigMaze

Iteration table (discount factor = 0.6)

Iterations (K)	Average Score	Win rate (%)
100	476.35	100
200	476.2	100
300	476.7	100
400	476.2	100
500	476.15	100

Since the game did not have the highest average score when K = 500, we can stop incrementing K, and chooses K = 300 as our iteration value, which has the highest average score among the five runs.

Discount factor table (iteration value = 300)

Discount factor (γ)	Average Score	Win rate (%)
0.3	476.3	100
0.35	476.15	100
0.4	475.95	100
0.45	473.7	100
0.5	475.95	100
0.55	475.3	100
0.6	474.8	100

0.65	475.4	100
0.7	476.3	100
0.75	476.5	100
0.8	476.4	100
0.85	476.5	100
0.9	476.45	100
0.95	476.15	100
1	474.55	100

Since the highest average score occur at both γ = 0.75 and γ = 0.85 (both are acceptable, but we will be choosing γ = 0.75), thus the final hyper-parameter will be γ = 0.75 and K = 300, giving an average score of 476.5 with a 100% win rate.

<u>bigMaze2</u>

Iteration table (discount factor = 0.6)

Iterations (K)	Average Score	Win rate (%)
100	479.35	100
200	481.25	100
300	481.6	100
400	480.15	100
500	482.6	100
600	480.3	100

Since the average score decreases at K = 600, we will then choose K = 500 as our iteration value, where the algorithm converges.

Discount factor table (iteration value = 500)

Discount factor (γ)	Average Score	Win rate (%)
0.3	480.35	100
0.35	480.9	100
0.4	479.35	100
0.45	481.6	100
0.5	480.7	100
0.55	482.7	100
0.6	480.5	100
0.65	480.7	100
0.7	481.05	100
0.75	480.9	100
0.8	480.95	100
0.85	481.05	100
0.9	481.45	100
0.95	479.6	100
1	480.45	100

Since the highest average score occur at γ = 0.55, thus the final hyper-parameter will be γ = 0.55 and K = 500, giving an average score of 482.7 with a 100% win rate.

contoursMaze

Iteration table (discount factor = 0.6)

Iterations (K)	Average Score	Win rate (%)
100	495.35	100
200	493.4	100
300	495.05	100
400	495.75	100
500	494.65	100

Since the game did not have the highest average score when K = 500, we can stop incrementing K, and chooses K = 400 as our iteration value, which has the highest average score among the five runs.

Discount factor table (iteration value = 400)

Discount factor (γ)	Average Score	Win rate (%)
0.3	493.9	100
0.35	494.3	100
0.4	494.15	100
0.45	494.6	100
0.5	493.95	100
0.55	494.65	100
0.6	494.6	100
0.65	493.5	100
0.7	493.5	100
0.75	494.7	100
0.8	494.15	100
0.85	493.35	100
0.9	493.55	100
0.95	494.05	100
1	494.3	100

Since the highest average score occur at γ = 0.75, thus the final hyper-parameter will be γ = 0.75 and K = 400, giving an average score of 494.7 with a 100% win rate

mediumMaze

Iteration table (discount factor = 0.6)

Iterations (K)	Average Score	Win rate (%)
100	439.05	95
200	337.45	85
300	287	80
400	387.75	90
500	287.2	80

Since the game did not have the highest average score when K = 500, we can stop incrementing K, and chooses K = 100 as our iteration value, which has the highest average score among the five runs.

Discount factor table (iteration value = 100)

Discount factor (γ)	Average Score	Win rate (%)
0.3	437.15	95
0.35	387.8	90
0.4	388.45	90
0.45	487.95	100
0.5	286.45	80
0.55	387.2	90
0.6	388.95	90
0.65	388.2	90
0.7	438.15	95
0.75	388.25	90
0.8	237.3	75
0.85	337.1	85
0.9	388.9	90
0.95	437.45	95
1	461.05	100

Since the highest average score occur at both γ = 0.3 and γ = 0.45 (both are acceptable, but we will be choosing γ = 0.45), thus the final hyper-parameter will be γ = 0.45 and K = 100, giving an average score of 487.95 with a 100% win rate.

<u>mediumMaze2</u>

Iteration table (discount factor = 1)

Iterations (K)	Average Score	Win rate (%)
100	482.4	100
200	433.55	95
300	483.25	100
400	482.95	100
500	483.75	100
600	484.1	100
700	484.3	100
800	484.75	100
900	483.2	100

Due to the discount factor being too low for this particular maze, the pacman will have a very low win rate, therefore we will be using a discount factor of 1 to find the K.

Since the average score decreases at K = 900, we will then choose K = 800 as our iteration value, where the algorithm converges.

Discount factor table (iteration value = 800)

Discount factor (γ)	Average Score	Win rate (%)
0.5	242.1	75
0.55	443.75	95
0.6	443.3	95
0.65	91.65	60
0.7	343.9	85
0.75	41.95	55
0.8	191.95	70
0.85	190.75	70
0.9	241.4	75
0.95	435.35	95
1	484.45	100

The pacman could not find a terminating path between γ = 0.3 and γ = 0.45, which is too low for this particular maze. Therefore, we will disregard the results obtained between γ = 0.3 and γ = 0.45.

Due to the low discount factors (between 0.5 and 0.95), the pacman will always chooses the shortest but dangerous path instead of the safest one, leading to an inconsistent win rate. Thus, the best choice the discount factor here is 1, giving an average score of 484.45 with a 100% win rate.

<u>openMaze</u>

Iteration table (discount factor = 0.6)

Iterations (K)	Average Score	Win rate (%)
100	440.25	100
200	439.75	100
300	440.05	100
400	438.55	100
500	438.65	100

Due to the discount factor being too low for this particular maze, the pacman will take a very long time to find a terminating path, therefore we will be using a discount factor of 1 to find the K.

Since the game did not have the highest average score when K = 500, we can stop incrementing K, and chooses K = 100 as our iteration value, which has the highest average score among the five runs.

Discount factor table (iteration value = 100)

Discount factor (γ)	Average Score	Win rate (%)
0.9	325.65	85
0.95	278	80
1	439.85	100

Due to the high discount factor requirement for this particular maze, the γ-value between 0.3 and 0.85 are too low, resulting in the pacman taking a very long time to terminate. Therefore, we will only consider the γ-values between 0.9 and 1.

Since the highest average score occur at both γ = 1, thus the final hyper-parameter will be γ = 1 and K = 100, giving an average score of 439.85 with a 100% win rate.

smallMaze

Iteration table (discount factor = 0.6)

Iterations (K)	Average Score	Win rate (%)
100	476.8	100
200	475.75	100
300	477.35	100
400	475.65	100
500	477.15	100

Since the game did not have the highest average score when K = 500, we can stop incrementing K, and chooses K = 300 as our iteration value, which has the highest average score among the five runs.

Discount factor table (iteration value = 300)

Discount factor (γ)	Average Score	Win rate (%)
0.3	476.7	100
0.35	475.75	100
0.4	477.15	100
0.45	477.95	100
0.5	476.8	100
0.55	476.85	100
0.6	476.05	100
0.65	476.05	100
0.7	474.95	100
0.75	214.7	75
0.8	431.75	95
0.85	384.05	90
0.9	334.55	85
0.95	335.5	85
1	474.75	100

Since the highest average score occur at γ = 0.45, thus the final hyper-parameter will be γ = 0.45 and K = 300, giving an average score of 477.95 with a 100% win rate.

testMaze

Iteration table (discount factor = 0.6)

Iterations (K)	Average Score	Win rate (%)
100	476.75	100
200	476.15	100
300	476.8	100
400	475.75	100
500	477.05	100
600	475.9	100

Since the average score decreases at K = 600, we will then choose K = 500 as our iteration value, where the algorithm converges

Discount factor table (iteration value = 500)

Discount factor (γ)	Average Score	Win rate (%)
0.3	475.9	100
0.35	476.1	100
0.4	476.5	100
0.45	476.15	100
0.5	476.5	100
0.55	476.4	100
0.6	476.05	100
0.65	475.6	100
0.7	476.7	100
0.75	476.55	100
0.8	476.35	100
0.85	476.05	100
0.9	476.45	100
0.95	477.8	100
1	476.15	100

Since the highest average score occur at γ = 0.95, thus the final hyper-parameter will be γ = 0.95 and K = 500, giving an average score of 477.8 with a 100% win rate.

tinyMaze

Iteration table (discount factor = 0.6)

Iterations (K)	Average Score	Win rate (%)
100	498.55	100
200	498.8	100
300	498.45	100
400	498.3	100
500	499.75	100
600	498.75	100

Since the average score decreases at K = 600, we will then choose K = 500 as our iteration value, where the algorithm converges

Discount factor table (iteration value = 500)

Discount factor (γ)	Average Score	Win rate (%)
0.3	499.05	100
0.35	498.4	100
0.4	499.25	100
0.45	498.65	100
0.5	497.6	100
0.55	498.65	100
0.6	499	100
0.65	499.35	100
0.7	498.9	100
0.75	499.8	100
0.8	498.85	100
0.85	498.85	100

0.9	499.8	100
0.95	499.5	100
1	498.6	100

Since the highest average score occur at $\gamma = 0.9$, thus the final hyper-parameter will be $\gamma = 0.9$ and K = 500, giving an average score of 499.8 with a 100% win rate.

trickyMaze

Iteration table (discount factor = 0.6)

Iterations (K)	Average Score	Win rate (%)
100	399.2	80
200	499.95	95
300	398.95	90
400	248.1	75
500	500.05	100
600	399.65	90

Since the average score decreases at K = 600, we will then choose K = 500 as our iteration value, where the algorithm converges

Discount factor table (iteration value = 500)

Discount factor (γ)	Average Score	Win rate (%)
0.3	476.95	100
0.35	478.3	100
0.4	477.4	100
0.45	450	95
0.5	348.85	85
0.55	399.4	90
0.6	399.45	90
0.65	399.55	90
0.7	449.9	95
0.75	349.2	85
0.8	348.95	85
0.85	449.85	95
0.9	398.95	90
0.95	450.15	95
1	478.15	100

Since the highest average score occur at $\gamma = 0.3$, thus the final hyper-parameter will be $\gamma = 0.3$ and K = 500, giving an average score of 478.3 with a 100% win rate.

Task 2b (Q-learning with epsilon greedy)

<u>Approach used</u>

Firstly, we will determine the discount factor (γ) and the learning rate (α). We will conduct a series of runs to identify the γ-value and α-value that yields the highest average score for each maze. We will test the γ-values in the range from 0.5 to 1, with an incremental step of 0.1, and the α-values in the range from 0.2 to 0.6, also with an incremental step of 0.1. Throughout these runs to determine the optimal γ-values and α-values, we will keep the default epsilon value (ε = 0.2) and use the default number of training iterations (K = 200).

Once we have determined the best γ-value and α-value for each maze, then we will determine the number of training iterations for each maze. We will start with 100 iterations and increment it by 100 each run until it converges. Convergence is defined when the average score over the last 100 episodes shows no significant improvement (<20 points of improvements). During these runs, we will maintain the default epsilon value (ε = 0.2).

Lastly, we will determine the epsilon value, taking into account the selected γ-value, α-value and the number of training iterations for each maze. The epsilon value will start from 0.2 and gradually decrease to 0.9, with an incremental step of 0.1. During these runs, if there is a significant deterioration shown over the average score, then the run will terminate, and the optimal epsilon value will be returned.

<u>bigMaze</u>

Discount factor (γ) & Learning rate (α) table (ε = 0.2, K = 200)

	α					
γ		0.2	0.3	0.4	0.5	0.6
	0.5	269.15	329.88	330.27	363.67	384.3
	0.6	282.18	314.08	344.69	361.01	385.77
	0.7	306.55	374.88	366.11	386.73	382.86
	0.8	273.87	309.81	345.43	374.89	375.98
	0.9	316.41	341.53	377.38	398.51	357.1
	1	325.16	336.85	378.7	373.77	389.02

Since the highest average score over all training occurs at γ = 0.9 and α = 0.5, they will be the optimal selection for our discount factor and learning rate respectively, yielding a final score of 480.

Iteration table (γ = 0.9, α = 0.5 and ε = 0.2)

Iterations	Average score over the last 100 iterations
100	329.23
200	471.4
300	471.4

Since the average score did not show significant improvement between 200 iterations and 300 iterations, thus we will choose 200 iterations as our K-value.

Epsilon (ε) table (γ = 0.9, α = 0.5 and K = 200)

ε	Average score over all trainings
0.2	369.47
0.3	351.63
0.4	315.71

Since the average score over all trainings drops off significantly at ε = 0.4, therefore we will choose 0.3 as our optimal ε-value. Thus, our final optimal values are ε = 0.3, γ = 0.9, α = 0.5 and K = 200.

bigMaze2

Discount factor (γ) & Learning rate (α) table (ε = 0.2, K = 200)

	α					
γ		0.2	0.3	0.4	0.5	0.6
	0.5	322.69	369.79	336.36	344.07	346.25
	0.6	359.6	358.14	334.91	359.74	372.51
	0.7	356.94	366.93	381.25	379.39	360.12
	0.8	369.06	371.4	386.6	326.01	390.57
	0.9	345.94	354.4	364.67	365.72	388.72
	1	309.2	373.94	345.01	392.49	352.69

Since the highest average score over all training occurs at γ = 1 and α = 0.5, they will be the optimal selection for our discount factor and learning rate respectively, yielding a final score of 489.

Iteration table (γ = 1, α = 0.5 and ε = 0.2)

Iterations	Average score over the last 100 iterations
100	351.59
200	403.76
300	433.03
400	404.26

Since the average score did not show significant improvement between 300 iterations and 400 iterations, thus we will choose 300 iterations as our K-value.

Epsilon (ε) table (γ = 1, α = 0.5 and K = 300)

ε	Average score over all trainings
0.2	397.57
0.3	385.43
0.4	314.78

Since the average score over all trainings drops off significantly at ε = 0.4, therefore we will choose 0.3 as our optimal ε-value. Thus, our final optimal values are ε = 0.3, γ = 1, α = 0.5 and K = 300.

contoursMaze

Discount factor (γ) & Learning rate (α) table (ε = 0.2, K = 200)

	α					
γ		0.2	0.3	0.4	0.5	0.6
	0.5	477.14	477.54	481.74	482.32	483.37
	0.6	476.24	484.41	481.56	480.9	482.6
	0.7	479.78	479.55	482.89	482.56	481.54
	0.8	480.02	481.23	480.65	481.76	484.44
	0.9	478.29	480.27	481.68	483.32	485.06
	1	479.56	480.66	483.05	484.21	483.66

Since the highest average score over all training occurs at γ = 0.9 and α = 0.6, they will be the optimal selection for our discount factor and learning rate respectively, yielding a final score of 497.

Iteration table (γ = 0.9, α = 0.6 and ε = 0.2)

Iterations	Average score over the last 100 iterations
100	471.34
200	493.83
300	493.57

Since the average score did not show significant improvement between 200 iterations and 300 iterations, thus we will choose 200 iterations as our K-value.

Epsilon (ε) table (γ = 0.9, α = 0.6 and K = 200)

ε	Average score over all trainings
0.2	485.46
0.3	481.62
0.4	476.68
0.5	472.96
0.6	464.41
0.7	456.33
0.8	415.73

Since the average score over all trainings drops off significantly at ε = 0.8, therefore we will choose 0.7 as our optimal ε-value. Thus, our final optimal values are ε = 0.7, γ = 0.9, α = 0.6 and K = 200.

mediumMaze

Discount factor (γ) & Learning rate (α) table (ε = 0.2, K = 200)

	α					
γ		0.2	0.3	0.4	0.5	0.6
	0.5	182.24	198.12	273.46	294.49	293.07
	0.6	180.25	204.98	282.82	252.69	263.89
	0.7	182.15	222.21	274.07	285.19	329.62
	0.8	151.96	252.41	225.51	299.46	291.95
	0.9	229.32	226.08	226.27	245.36	275.93
	1	214.56	201.5	241.69	231.92	263.69

Since the highest average score over all training occurs at γ = 0.7 and α = 0.6, they will be the optimal selection for our discount factor and learning rate respectively, yielding a final score of 487.

Iteration table (γ = 0.7, α = 0.6 and ε = 0.2)

Iterations	Average score over the last 100 iterations
100	168.76
200	469.5
300	481.64

Since the average score did not show significant improvement between 200 iterations and 300 iterations, thus we will choose 200 iterations as our K-value.

Epsilon (ε) table (γ = 0.7, α = 0.7 and K = 200)

ε	Average score over all trainings
0.2	285.14
0.3	248.31

Since the average score over all trainings drops off significantly at ε = 0.3, therefore we will choose 0.2 as our optimal ε-value. Thus, our final optimal values are ε = 0.2, γ = 0.7, α = 0.6 and K = 200.

openMaze

Discount factor (γ) & Learning rate (α) table (ε = 0.2, K = 200)

	α					
γ		0.2	0.3	0.4	0.5	0.6
	0.5	-117.62	-143.29	-15.23	-26.09	-70.98
	0.6	-124.25	-122.26	-226.66	-38.34	-37.88
	0.7	-163.09	-98.2	-80.29	-67.44	-54.79
	0.8	-190.02	-171.51	-145.5	-64.58	-112.45
	0.9	-152.89	-70.73	-173.06	-33.33	-96.08
	1	-90.47	-83.75	-95.06	-53.26	-144.78

Since the highest average score over all training occurs at γ = 0.5 and α = 0.4, they will be the optimal selection for our discount factor and learning rate respectively, yielding a final score of 500.

Iteration table (γ = 0.5, α = 0.4 and ε = 0.2)

Iterations	Average score over the last 100 iterations
100	-422.09
200	182.25
300	176.26

Since the average score did not show significant improvement between 200 iterations and 300 iterations, thus we will choose 200 iterations as our K-value.

Epsilon (ε) table (γ = 0.5, α = 0.4 and K = 200)

ε	Average score over all trainings
0.2	-68.26
0.3	-167.36

Since the average score over all trainings drops off significantly at ε = 0.3, therefore we will choose 0.2 as our optimal ε-value. Thus, our final optimal values are ε = 0.2, γ = 0.5, α = 0.4 and K = 200.

smallMaze

Discount factor (γ) & Learning rate (α) table (ε = 0.2, K = 200)

	α					
γ		0.2	0.3	0.4	0.5	0.6
	0.5	270.81	260.57	316.79	260.4	315.88
	0.6	266.8	302.04	289.6	272.25	353.48
	0.7	299.25	288.05	319.46	290.56	323.36
	0.8	265.11	288.48	288.36	325.88	321.38
	0.9	259.99	307.38	304.39	308.74	305.48
	1	306.35	288.19	336.24	306.79	306.83

Since the highest average score over all training occurs at γ = 0.6 and α = 0.6, they will be the optimal selection for our discount factor and learning rate respectively, yielding a final score of 495.

Iteration table (γ = 0.6, α = 0.6 and ε = 0.2)

Iterations	Average score over the last 100 iterations
100	201.07
200	401.07
300	361.12

Since the average score did not show significant improvement between 200 iterations and 300 iterations, thus we will choose 200 iterations as our K-value.

Epsilon (ε) table (γ = 0.6, α = 0.6 and K = 200)

ε	Average score over all trainings
0.2	290.32
0.3	249.81
0.4	111.2

Since the average score over all trainings drops off significantly at ε = 0.3, therefore we will choose 0.2 as our optimal ε-value. Thus, our final optimal values are ε = 0.2, γ = 0.6, α = 0.6 and K = 200.

smallMaze2

Discount factor (γ) & Learning rate (α) table (ε = 0.2, K = 200)

	α					
γ		0.2	0.3	0.4	0.5	0.6
	0.5	148.5	223.21	196.94	270.25	250.79
	0.6	210.78	218.84	243.82	274.84	307.54
	0.7	178.19	195.44	253.69	267.39	292.95
	0.8	210.03	226.45	210.78	266.13	286.29
	0.9	159.47	261.52	295.73	312.83	303.12
	1	186.85	287.88	292.32	247.6	239.03

Since the highest average score over all training occurs at γ = 0.9 and α = 0.5, they will be the optimal selection for our discount factor and learning rate respectively, yielding a final score of 490.

Iteration table (γ = 0.9, α = 0.5 and ε = 0.2)

Iterations	Average score over the last 100 iterations
100	139.71
200	442.71
300	422.23

Since the average score did not show significant improvement between 200 iterations and 300 iterations, thus we will choose 200 iterations as our K-value.

Epsilon (ε) table (γ = 0.9, α = 0.5 and K = 200)

ε	Average score over all trainings
0.2	252.4
0.3	244.81
0.4	92.96

Since the average score over all trainings drops off significantly at ε = 0.4, therefore we will choose 0.3 as our optimal ε-value. Thus, our final optimal values are ε = 0.3, γ = 0.9, α = 0.5 and K = 200.

testMaze

Discount factor (γ) & Learning rate (α) table (ε = 0.2, K = 200)

	α					
γ		0.2	0.3	0.4	0.5	0.6
	0.5	456.33	463.33	466.46	467.94	469.36
	0.6	458.76	464.64	467.12	468.82	470.53
	0.7	460.23	464.9	467.46	468.52	470.18
	0.8	461	465.88	468.04	469.64	470.83
	0.9	462.68	467.06	468.93	470.18	470.18
	1	463.25	466.92	468.85	470.18	470.31

Since the highest average score over all training occurs at γ = 0.8 and α = 0.6, they will be the optimal selection for our discount factor and learning rate respectively, yielding a final score of 483.

Iteration table (γ = 0.8, α = 0.5 and ε = 0.2)

Iterations	Average score over the last 100 iterations
100	465.16
200	476.56

Since the average score did not show significant improvement between 100 iterations and 200 iterations, thus we will choose 100 iterations as our K-value.

Epsilon (ε) table (γ = 0.8, α = 0.5 and K = 100)

ε	Average score over all trainings
0.2	463.02
0.3	458.82
0.4	452.55
0.5	443.22
0.6	432.31

Since the average score over all trainings drops off significantly at ε = 0.6, therefore we will choose 0.5 as our optimal ε-value. Thus, our final optimal values are ε = 0.5, γ = 0.8, α = 0.5 and K = 100.

tinyMaze

Discount factor (γ) & Learning rate (α) table (ε = 0.2, K = 200)

	α					
γ		0.2	0.3	0.4	0.5	0.6
	0.5	405.91	427.2	442.99	407.51	402.45
	0.6	416.37	412.2	412.46	402.81	412.75
	0.7	431.93	402.08	422.59	417.93	427.78
	0.8	381.22	427.69	442.84	402.63	397.58
	0.9	401.76	407.14	427.34	407.87	408
	1	386.77	432.63	422.94	392.94	433.48

Since the highest average score over all training occurs at γ = 0.5 and α = 0.4, they will be the optimal selection for our discount factor and learning rate respectively, yielding a final score of 502.

Iteration table (γ = 0.5, α = 0.4 and ε = 0.2)

Iterations	Average score over the last 100 iterations
100	415.94
200	409.14

Since the average score did not show significant improvement between 100 iterations and 200 iterations, thus we will choose 100 iterations as our K-value.

Epsilon (ε) table (γ = 0.5, α = 0.4 and K = 100)

ε	Average score over all trainings
0.2	395.83
0.3	344.06

Since the average score over all trainings drops off significantly at ε = 0.3, therefore we will choose 0.2 as our optimal ε-value. Thus, our final optimal values are ε = 0.2, γ = 0.5, α = 0.4 and K = 100.

<u>trickyMaze</u>

Discount factor (γ) & Learning rate (α) table (ε = 0.2, K = 200)

	α					
γ		0.2	0.3	0.4	0.5	0.6
	0.5	334.97	339.68	365.46	332.76	340.22
	0.6	314.29	301.94	372.05	373.55	360.95
	0.7	349.92	348.5	336.19	354.38	321
	0.8	322.11	302.98	348.79	329.67	345.94
	0.9	323.47	351.2	293.37	320.62	357.57
	1	364.35	355.68	334.52	310.5	362.27

Since the highest average score over all training occurs at γ = 0.6 and α = 0.5, they will be the optimal selection for our discount factor and learning rate respectively, yielding a final score of 493.

Iteration table (γ = 0.6, α = 0.5 and ε = 0.2)

Iterations	Average score over the last 100 iterations
100	280.57
200	378.72
300	379.07

Since the average score did not show significant improvement between 200 iterations and 300 iterations, thus we will choose 200 iterations as our K-value.

Epsilon (ε) table (γ = 0.6, α = 0.5 and K = 200)

ε	Average score over all trainings
0.2	323.35
0.3	270.96

Since the average score over all trainings drops off significantly at ε = 0.3, therefore we will choose 0.2 as our optimal ε-value. Thus, our final optimal values are ε = 0.2, γ = 0.6, α = 0.5 and K = 200.

Part 2 – Machine Learning

Task 3 (Single Layer Perceptron)

<u>Classification vs Regression</u>

The objective of this task is to design and train a single-layer perceptron model that predicts the best legal action for a given input feature, which is a model that has the ability to make categorical decisions, rather than predicting a continuous value. In this context, the model's role is to classify all the legal actions into two categories: favourable and unfavourable. Therefore, this would be a classification task rather than a regression task.

<u>Activation function and loss function</u>

Sigmoid function (Activation function)
The sigmoid function produces outputs between 0 and 1, which can be useful for quantifying the model's confidence in its predictions. For instance, a value close to 1 can indicate a high confidence of predicting an action to be favourable, and a value close to 0 can indicate a high confidence of predicting an action to be unfavourable. While a value around 0.5 can indicate that the model is unsure whether the action is favourable or unfavourable.

Binary Cross Entropy (Loss function)

The cross entropy quantifies the difference between two probability distributions. In our case, the model will predict a binary distribution {p, 1-p}, where 'p' is the probability of the favourable action class, and '1-p' is the probability of the unfavourable action class. We can then use the binary cross entropy to compare it with the true distribution {y, 1-y}, where 'y' is the true value of the favourable action class, and '1-y' is the true value of the unfavourable action class.

<u>Approach used to determine number of training iterations and learning rate (α)</u>

To determine the optimal number of training iterations and learning rate, we will conduct a series of tests with varying values. For each iteration, ranging from 20 to 200 in increments of 20, we will explore learning rates between 0.2 and 1, incremented by 0.1. During each test, we will employ these values to train the model and then use the trained model to play the game 5 times, specifically using the 'mediumClassic' maze for these tests. Subsequently, we will calculate the average score of these games to evaluate the performance for each value.

mediumClassic

	Number of training iterations										
α		20	40	60	80	100	120	140	160	180	200
	0.2	292.6	341.4	367	174	481.8	291.4	136.2	189.8	200.6	254.3
	0.3	156.6	567	214	35.6	129	472.8	338.2	240.6	280.8	181
	0.4	374.6	10.4	197.6	124.2	170.2	224.8	550	575.6	508.2	178
	0.5	455.4	606.8	353.4	237.8	391.2	366.4	208.8	415.4	312	174.2
	0.6	279.8	484.8	810.8	660.2	151.2	133	261.8	254.2	199.2	271
	0.7	189.4	240.8	488.6	250.8	34.8	96.4	420.8	176.6	628.2	62.4
	0.8	212.6	228.2	207	140.2	193.6	420.6	385.6	102.4	319.6	422.6
	0.9	257	266.6	574.8	300.6	442.8	224	185	201	360	280
	1	291.8	125.4	495.4	670.4	85.2	26.2	204.2	303.2	185.2	547.8

Learning rate table

Learning rate (α)	Average score for each α
0.2	272.91
0.3	261.56
0.4	291.36
0.5	352.14
0.6	350.6
0.7	258.88
0.8	263.24
0.9	309.18
1	293.48

Since 0.5 learning rate has the highest average score, it will be the optimal selection for our learning rate.

Number of training iterations table

Number of training iterations	Average score
20	278.87
40	319.04
60	412.07
80	288.2
100	231.09
120	250.62
140	298.96
160	273.2
180	332.64
200	263.48

Based on the table above, training the model for 60 times returns the highest average score, therefore it will be our optimal number of training iterations.

Thus, our final optimal values are α = 0.5 and number of training iterations = 60.

Acknowledgments

- “Deep Learning: Which loss and Activation Functions should I use?” by Stacey Ronaghan, July 2018: provided inspiration and guidance for part 2 of the assignment.
- “Derivation of the Binary Cross Entropy Loss Gradient” by Gianluca Turcatel, December 2021: provided inspiration and guidance for part 2 of the assignment.