MLPR SEMESTER 4 PROJECT

Dipit Golechha, Harsh Siroya, **Krishnav Mahansaria**



Rende Ining Raving Bleed of the single sin



What's the Problem?

Blackjack combines skill, strategy, and luck, challenging players with the critical choice of 'hit', 'stand', 'double down' and 'surrender'. Our project aims to automate these decisions using a reinforcement learning model to navigate the game's complexity and potential outcomes, aiming to outperform traditional strategies.

Why This Problem?

This will not only allow us to make playing blackjack easier and increase the odds of winning but will also allow us to explore different games and problems which require similar strategy making and consists of similar probabilistic outcome allowing us to reimagine the use of RL.



Goal

We have developed a stratergy using a sophisticated reinforcement learning model designed to minimise risk in Blackjack.



Literature Review

Previously, we had researched on the methods we can use and came across some papers telling us about the methods' effectiveness in playing blackjack. One study explored Deep Q-learning, comparing Deep Q-Network (DQN) models against a traditional Q-Network (QN) model. The DQN models outperformed the QN model, indicating promise in learning effective strategies for blackjack. However, none of the models discovered the exact optimal strategy, suggesting room for improvement.

Another paper investigated the application of the Q-learning algorithm, showcasing its potential in approximating an optimal blackjack strategy. Despite not achieving perfect convergence, the reinforcement learning (RL) agent showed significant improvement over random actions, approaching the performance of a basic strategy player.

> "Learning to Play Blackjack with Deep Learning and Reinforcement Learning" by Ish Handa "Applying Reinforcement Learning to Blackjack Using Q-Learning" by Charles de Granville "Playing Blackjack with Deep Q-Learning" by Allen Wu from Stanford University

"Learning to Play Blackjack with Deep Learning and Reinforcement Learning" by Ish Handa



(a) using random strategy

Figure 6: Average Payoff after 10000 hands per round to estimate range

(b) using Edward Thorp's strategy

"Applying Reinforcement Learning to Blackjack Using Q-Learning" by Charles de Granville



"Playing Blackjack with Deep Q-Learning" by Allen Wu from Stanford University

player/deale	r .	2	3	4	5	6	7	8	9 1	IO A	player/dea	ler	2	3	4	5	6	7	8	9 1	10 A
Solution Material State	5 hit	hit	hit	hit	hit	hit	hit	hit	hit	hit	Accessible of Association	5 hit	hit	hit	NR	hit	hit	hit	hit	hit	hit
	6 hit	hit	hit	hit	hit	hit	hit	hit	hit	hit		6 hit	hit	hit	hit	hit	nit	bit	hit	hit	hit
	7 hit	hit	hit	hit	hit	hìt	hít	hit	hit	thit		7 hit	hit	hit	bit	hit	hit	hit	hit	hit	hit
	8 hit	hit	hit	hit	hit	hit	hit	hit	hit	hit		8 hit	hit	hit	hit	hit	hit	hit	hit	hit	hit
	9 hit	double	double	double	double	hit	hit	hit	bit	hit		9 double	double	double	double	double	hit	hit	hit	hit	hit
1	0 double	double	double -	double	double	double	double	double	hit	hit		10 double	double	double	double	double	double	double	hit	hit	hit
1	1 double	double	double	double	double	double	double	double	double	double		11 double	double	double	double	double	double	double	double	hit	hit
1	2 hit	hit	stand	stand	stand	hit	hit	hit	hit	hit		12 double	hit	hit	hit	hit	hit	hit	hit	hit	hit
1	3 stand	stand	stand	stand	stand	hit	hit	hit	hit	hit		13 hit	hit	hit	hit	hit	hit	bit	hit	hit	hit
1	4 stand	stand	stand	stand	stand	hit	hit	hit	hit	hit		14 double	hit	hit	hit	hit	hit	hit	hit	hit	surrender
1	5 stand	stand	stand	stand	stand	hit	hit	hit	surrender	hit		15 hit	hit	stand	stand	hit	hit	hit	bit	surrender	surrender
1	6 stand	stand	stand	stand	stand	hit	bit	summder	surrender	surrender		16 hit	hit	hit	stand	stand	stand	surronder	surrender	surrender	surronder
1	7 stand	stand	stand	stand	stand	stand	stand	stand	stand	stand		17 stand	stand	stand	stand	stand	stand	stand	stand	surronder	surrender
1	8 stand	stand	stand	stand	stand	stand	stand	stand	stand	stand		18 stand	stand	stand	stand	stand	stand	stand	stand	stand	turrender
1	9 stand	stand	stand	stand	stand	stand	stand	stand	stand	stand		19 stand	stand	stand	stand	stand	stand	stand	stand	stand	stand
2	0 stand	stand	stand	stand	stand	stand	stand	stand	stand	stand		20 stand	stand	stand	stand	stand	stand	stand	stand	stand	stand
soft 13	hit	hit	hit	double	double	bit	hit	hit	bit	hit	soft 13	hit	hit	hit	hit	hit	hit	hit	hit	desible	hit
soft 14	hit	hit	hit	double	double	hit	hit	hit	hit	hit	soft 14	hit	hit	bit	hit	hit	hit	hit	hit	hit	hit
soft 15	hit	hit	dauble	double	double	hit	hit	hit	hit	bút	soft 15	hit	hit	hit	hit	hit	hit	this	hit	hit	hit
soft 16	hit	hit	double	double	double	hit	hit	hit	hit	hit	soft 16	hit	hit	hit	hit	hit	hit	hit	hit	hit	hit
soft 17	hit	double	double	double	double	hit	hit	hit	hit	hit	soft 17	double	double	double	hit	hit	hit	Pilt	hit	hit	hit
soft 18	double	double	double	double	double	stand	stand	Bit	hit	hit	soft 18	stand	stand	stand	double	double	stand	stand	stand	hit	Purt
soft 19	stand	stand	stand	stand	double	stand	stand	stand	stand	stand	soft 19	stand	stand	stand	stand	stand	stand	stand	stand	stand	stand
soft 20	stand	stand	stand	stand	stand	stand	stand	stand	stand	stand	soft 20	stand	stand	stand	stand	stand	stand	stand	stand	stand	stand

(a) Optimal policy [4]

(b) Derived policy

"Playing Blackjack with Deep Q-Learning" by Allen Wu from Stanford University

Network	Policy Score	Similarit
QN	228.5	0.6720588
DQN(3,1)	248.5	0.7308823
DQN(13,7)	249.5	0.7338235
DQN(11, 13)	259.5	0.7632352
optimal policy	340	1

DQN(depth (no of layers), width (number of neurons))



Why we chose RL over supervised and unsupervised?

Reinforcement Learning (RL) stands out as the preferred approach for training a blackjack-playing model due to its ability to optimize rewards, adapt to dynamic gameplay, balance risk and reward, learn from game outcomes, and handle uncertainty. Unlike supervised learning, which relies on predetermined optimal moves, and unsupervised learning, which lacks explicit reward signals.

- We're implementing a reinforcement learning (RL) model using the **OpenAl Gym Blackjack environment accessed via API calls. This** environment simulates the classic card game, providing a framework for our RL agent to learn and improve its strategies through interactions with the game.
- Since we are using RL and not ML, the model has to learn through its actions and hence has to be given the freedom of action that is either exploitation or exploration. Whereas if the model was trained using a datset it would have been limited to the decisions and action mentioned in the dataset.

Datasei

- Through the OPen AI Gym Model we were able to get the following state space: **Players Sum Dealers Up Card Usable Ace Burst or Not** Reward
- Following action space: Hit, Stand, Double Down
- Number of rounds of blackjack played 20 Million



Double Q Learning

Double Q-learning is an extension of Q-learning that addresses the issue of overestimation of action values. In traditional Q-learning, a single Q-value is updated based on the maximum Q-value for the next state which can lead to overestimation.

In Double Q-learning, two sets of Q-values are maintained, and during the update step, one set is used to select the best action, while the other set is used to evaluate the value of that action. This helps mitigate the overestimation problem by decoupling action selection from action evaluation.



Hyperparameters

- Alpha (α): Alpha is the learning rate, determining the extent to which new information overrides old information during Q-value updates. It typically ranges between 0 and 1, with higher values indicating more weight given to new experiences.
- Gamma (y): Gamma represents the discount factor, indicating the importance of future rewards relative to immediate rewards. It ranges between 0 and 1, with higher values prioritizing long-term rewards.
- Epsilon (ε): Epsilon is the exploration rate, governing the balance between exploration and exploitation in the agent's behavior. It determines the probability of selecting a random action instead of the optimal action according to the current policy.
- Number of Episodes: The number of episodes refers to the total number of training iterations or games played by the model. Each episode consists of interactions with the environment, updating Q-values based on observed rewards and transitions. The

Hyperparameters Optimisation

- We experimented with different values for gamma, alpha, and epsilon simultaneously in our Double Q-learning code.
- On running the code with all combinations, we evaluated the results to determine which combination of gamma, alpha, and epsilon produces the most optimal outcomes. The best-performing parameter values were then selected and incorporated into our model permanently.
- The best values were Alpha 0.1, Epsilon 0.2 and Gamma -**0.8**

Our Models Value Functions

- 1.5

- 1.0

- 0.5

- 0.0

77	Value Function with Usable Ace = Yes (Usable Ace = Yes)												
21 -	-0.01	0.52	0.62	0.73	0.58	0.81	0.71	0.72	0.20	0.44			
13	-0.02	0.75	0.58	0.30	0.61	0.54	0.46	0.39	0.05	0.39			
- 14	-0.09	0.49	0.67	0.56	0.85	0.78	0.25	0.57	0.21	0.07			
- 15	-0.07	0.27	0.20	0.37	0.50	0.64	0.84	0.09	-0.29	0.26			
s Sum 16	-0.43	0.19	0.28	0.20	0.19	0.99	0.31	0.92	0.14	0.10			
Player' 17	-0.22	0.36	0.10	0.36	0.47	0.74	0.18	-0.02	-0.21	-0.18			
- 18	-0.28	0.47	0.51	0.54	0.52	0.69	0.71	0.39	0.03	-0.03			
eI -	-0.14	0.86	0.51	1.11	0.42	0.46	1.49	1.50	0.28	0.16			
20	-0.23	1.01	0.28	1.86	0.79	1.86	1.11	1.59	1.47	0.81			
21	1.05	2.00	2.00	2.00	1.95	2.00	2.00	2.00	1.87	1.71			
	i	2	3	4	5 Dealer's Sh	6 owing Card	7	8	9	10			

	Value Function with Usable Ace = No (Usable Ace = No)													
4 -	-0.23	0.29	0.38	0.57	0.49	0.54	0.29	0.05	0.07	-0.07				
- ت	-0.36	-0.41	0.28	0.33	0.53	0.43	0.19	-0.19	-0.00	-0.06		- 2.0		
- ص	-0.37	0.17	0.25	-0.00	0.24	0.37	0.51	-0.14	-0.07	-0.12				
7	-0.65	0.22	0.33	0.24	0.69	0.63	0.38	0.04	-0.08	-0.38				
<u></u> ∞ -	-0.42	0.26	0.38	0.46	0.43	0.83	0.57	-0.08	-0.18	0.05		- 1.5		
ი -	-0.28	0.24	0.50	2.37	0.42	0.55	0.39	0.40	-0.10	-0.17				
9 -	-0.01	0.71	0.59	0.79	0.55	1.46	0.41	0.94	0.78	0.35		- 1.0		
្ន	-0.12	0.79	0.79	0.78	0.99	0.99	0.86	0.06	0.39	0.26				
s Sun 12	-0.62	-0.60	-0.39	-0.10	-0.38	-0.02	-0.28	-0.38	-0.78	-0.72		- 0.5		
layer' 13	-0.30	-0.23	0.11	-0.20	-0.38	-0.65	-0.30	-0.37	-0.94	-1.03				
14 P	-1.31	-0.67	0.20	-0.93	-0.65	-0.37	-0.12	-0.70	-0.08	-0.51				
- 15	-0.53	-1.45	-0.70	0.07	-0.15	-0.95	-0.54	-0.65	-0.87	-0.94		- 0.0		
16	-0.99	0.47	-0.94	-1.06	-0.79	-0.64	-1.03	-0.82	-0.68	-0.46				
17	-1.31	-0.32	0.17	-0.60	0.43	-0.07	-0.15	-0.65	-0.37	-1.15		0.5		
- 18	-0.57	0.61	0.95	0.72	0.08	0.86	0.44	0.46	-0.32	0.01				
et -	-0.50	0.78	1.01	1.00	1.35	1.07	1.83	1.36	0.63	0.31		1 0		
20	0.13	1.07	1.56	1.74	1.83	1.61	1.36	1.82	1.71	1.44		1.0		
21	1.24	2.00	1.94	1.88	1.92	1.81	1.99	1.91	1.77	1.69				
	i	2	3	4	5 Deplorie Sk	6	7	8	9	10				

Dealer's Showing Card

Our Models Suggested Strategy

	Value Function with Usable Ace = No (Usable Ace = No)										jest netton .		Value I	Function wi	th Usable A	ice = Yes (U	sable Ace	= Yes)			
4 -	н	Н	Н	н	н	н	Н	Н	н	Н					ч	110		. ц.		DD	ш
- م	H	H	Н	Н	н	н	H	н	н	н	11		00		30			in		00	
- و	Н	Н	Н	н	H	H	н	Н	Н	Н	m -	- н	н	Ĥ	н	н	THE	н	н	Ĥ	н
► -	H	Н	Н	Н	н	Н	н	Н	н	Н	1										
∞ -	H	н	Пні	Ĥ	н	н	н	н	н	н	14	- Н	н	н	н	н	н	н	н	н	н
ი -	Н	н	н	DD	н	н	Н	Н	н	н	0.02042										
9 -	Н	DD	н	н	н	DD	Н	н	н	н	15	H	H	н	н	н	Ξ Η (н	H.	н	н
្ន-	H	Н	н	н	н	н	Н	н	н	н											
s Sum 12	Н	S	н	н	S	н	Н	н	н	н	Sum 16	- Н	н	н	Н	н	DD	Н	DD	H	Н
ayer' 13	Н	Н	Гн	н	н	н	Н	н	н	Н	ayer's										
14 - 14	Н	S	н	s	S	н	Н	н	н	н	Pla 17	- H	н	H	Н	н	H	Н	Н	н	H
- 15	S	н	S	s	s	н	Н	н	н	н	ω -	-	н	н	н	H	TERC	н	SHE S	ЭH	н
- 16	Н	S	S	н	s	DD	Н	S	н	s	1	1772									
17	Н	S	S	S	s	S	S	s	S	S	6[-	- H	s	Ĥ	S	н	н	S	S	S	н
- 18	S	S	s	s	S	S	S	s	S	S	4.01.009										
- 19	S	S	s	s	S	S	S	s	s	S	20	Н	S	S	S	н	S	S	s	S	S
- 20	s	S	s	s	S	S	S	s	s	S											
- 21	S	S	S	S	S	S	S	S	S	S	21	- S	S	S	S	S	S	S	S	S	S
10	i	2	3	4	5 Dealer's Sh	6 owing Card	ż	8	9	10		i	2	3	4	5	6	7	8	9	10
Dealer's Showing Card															Dealer's Sh	owing Card					

How did we evaluate the model.

- Finding the winnings and losses of basic strategy over 100K games
- Finding the winnings and losses of our implemented strategy over 100K games
- Comparing the accuracy and working of the model.
- Comparing the model policy score with basic startegy (optimal)

over 100K games ted strategy over

lel. rtegy (optimal)

Model Accuracy



Policy Score

- For Ace Matrix 48.5/100
- For Non Ace Matrix 154/180
- Total = 202.5/280
- 72.3%





Hoping that you would never gamble





Top	perform	ing hyp	erparamet	ers:					
	Gamma	Alpha	Epsilon	Wins	Losses	Cur			
3	0.80	0.10	0.20	42859.0	47866.0				
0	0.80	0.10	0.10	42808.0	48492.0				
26	0.85	0.10	0.15	42719.0	47991.0				
109	1.00	0.15	0.30	42654.0	48578.0				
5	0.80	0.15	0.10	42647.0	49039.0				
	winper	ecentga	e						
3		42.85	9						
Θ		42.80	8						
26		42.71	9						
109		42.65	4						
5		42.64	.7						
Best	hyperp	aramete	rs:						
Gamma	а			0.800					
Alpha	а			0.100					
Epsi	lon			0.200					
Wins			4285	42859.000					
Loss	es		4786	47866.000					
Cumu	lative	Winning	s -486	4.000					
winp	erecent	gae	4	2.859					
Name	: 3, dt	ype: fl	oat64						

imulative Winnings \ -4864.0 -5647.0 -5107.0 -5790.0 -6334.0