MathWorks FINANCE CONFERENCE 2023

## Multiperiod Goal-Based Wealth Management using Reinforcement Learning

October 11-12 | Online



Valerio Sperandeo, MathWorks





A Goal-based Wealth Management Problem



Solving the Problem using Reinforcement Learning



Why use Reinforcement Learning





A Goal-based Wealth Management Problem



Solving the Problem using Reinforcement Learning



Why use Reinforcement Learning



# In Goal-Based Wealth Management the asset allocation aims at achieving a specific objective









# A classic example of Goal-Based Wealth Management is the retirement problem







# In the retirement problem the asset allocation of the portfolio depends on the proximity to the retirement date





Source: Vanguard





A Goal-based Wealth Management Problem



Solving the Problem using Reinforcement Learning



Why use Reinforcement Learning

## What is Reinforcement Learning?



Type of machine learning that trains an 'agent' through trial & error interactions with an environment



9

## What is Reinforcement Learning?

Type of machine learning that trains an 'agent' through trial & error interactions with an environment

What we need to define:

- 1. Actions
- 2. Observations
- 3. Environment
- 4. Reward
- 5. Agent





EINFORCEMENT LEARNING ALGORITHM

# The actions at each rebalancing period are a finite set of portfolio weights on the efficient frontier



📣 MathWorks<sup>。</sup>





## The observations from the environment are the wealth level and the time period





## Wealth Level Continuous Variable



**Time Period** Discrete Variable

## The reward functions provide a compensation for achieving a goal or for following a path



📣 MathWorks



## **Deep Q-Network (DQN)** agent based on the action and observation spaces

_															
	📣 Deep Learning Ne	twork Analyz	zer									-			
	Analysis for dla Name: criticNet Analysis date: 11-	-May-2023	<b>usage</b> 14:44:31							1.4k	<b>6</b> bles layers	0 A warnings	0 () errors		
				AN	ANALYSIS RESULT										
				Nam	ne	Туре	Activations	Learnable P	roper						
		a input 1	7	1	inpu 2 feat	it_1 tures	Feature Input	2(C) × 1(B)	-						
				2	fc_1 30 ful	Ilv connected laver	Fully Connected	30(C) × 1(B)	Weights 30 Bias 30	) × 2 ) × 1					
1 input_1 2 features			Feature Input	3	relu_ ReLU	body	ReLU	30(C) × 1(B)	-						
				4	fc_bo 30 ful	ody illy connected layer	Fully Connected	30(C) × 1(B)	Weights 30 Bias 30	) × 30 ) × 1					
Wealth Lev	vel	• TC_1		5	body ReLU	y_output	ReLU	30(C) × 1(B)	-						
Time Perio	d	+		6	outp 15 ful	out illy connected layer	Fully Connected	15(C) × 1(B)	Weights 15 Bias 1	2 fc	1			Fully Connected	
		<ul> <li>relu_body</li> <li>fc_body</li> <li>body_outp</li> <li>output</li> </ul>	, out		<sup>6</sup>	Possible	e Portfo	Fully Connected		30 F	Hidde	n Un	its		
														1	3



POLICY

ACTION

## **Reinforcement Learning Toolbox provides a rich set of Built-in agents**

Agent	Туре	Action Space	
SARSA Agents	Value-Based	Discrete	]
Policy Gradient (PG) Agents (PG)	Policy-Based	Discrete or continuous	
Actor-Critic (AC) Agents (AC)	Actor-Critic	Discrete or continuous	- On-
Trust Region Policy Optimization (TRPO) Agents (TRPO)	Actor-Critic	Discrete or continuous	
Proximal Policy Optimization (PPO) Agents (PPO)	Actor-Critic	Discrete or continuous	J
Q-Learning Agents (Q)	Value-Based	Discrete	]
Deep Q-Network (DQN) Agents	Value-Based	Discrete	
Deep Deterministic Policy Gradient (DDPG) Agents	Actor-Critic	Continuous	
Twin-Delayed Deep Deterministic (TD3) Policy Gradient Agents (TD3)	Actor-Critic	Continuous	► Off-
Soft Actor-Critic (SAC) Agents (SAC)	Actor-Critic	Continuous	
Model-Based Policy Optimization (MBPO) Agents (MBPO)	Actor-Critic	Discrete or continuous	J

**On-Policy Built-In Agents** 

• Off-Policy Built-In Agents

### A MathWorks

 ACTION

# Any custom agent can be defined as a subclass of the rl.agent.CustomAgent class

- 1. Create a subclass from the class rl.agent.CustomAgent
- 2. Define the appropriate Agent properties
- 3. Define a constructor function
- 4. Add a critic and an actor (if needed)
- 5. Define required agent methods
  - getActionImpl
  - getActionWithExplorationImpl
  - learnImpl

classdef CustomGAgent < rl.agent.CustomAgent</pre>

properties % G[state, action] G = [0, 0]% N[state, action] N = [0, 0]% actions actions % rho the prior policy rho % k: param for adjusting beta. % total T steps т % epsilon for exploration epsilon % discount discount end

CORSERVATION CONSERVATION CO

MathWorks<sup>®</sup>

end

methods

% Constructor

obj.k = k;

obj.T = T;

function obj = CustomGAgent(num state, num action,...

% Call the abstract class constructor

obj.G = zeros(num\_state, num\_action);

obj.N = zeros(num\_state, num\_action);

% Define the observation and action spaces

obj = obj@rl.agent.CustomAgent();

% Set the G and N matrices

obj.actions = 1:num\_action;

obj.epsilon = epsilon;

obj.discount = discount;

obj.ObservationInfo\_ = obs\_info; obj.ActionInfo\_ = act\_info;

k, epsilon, obs\_info, act\_info, T, discount)

obj.rho = ones(num\_state, num\_action) ./ num\_action;





A Goal-based Wealth Management Problem



Solving the Problem using Reinforcement Learning



Why use Reinforcement Learning



# The optimal strategy is to get more aggressive when the wealth is low, and as we are closer to the end of the investment horizon





					Act	ion					45
23.281	15	15	13	13	13	13	13	13	13	13	15
28.273	15	15	13	13	13	13	13	13	13	13	
34.336	15	15	13	13	13	13	13	13	13	13	445
41.7	15	15	15	13	13	13	13	13	13	13	14.5
50.642	15	15	15	15	13	13	13	13	13	13	
61.502	15	15	15	15	13	13	13	13	13	13	
74.692	15	15	15	15	15	13	13	13	13	13	- 14
90.709	15	15	15	15	15	15	15	13	13	13	
110.16	11	15	15	15	15	15	15	15	13	13	1.0 -
<u></u> 133.79	11	11	11	15	15	15	15	15	15	15	13.5
≥ 162.48	11	11	11	11	11	15	15	15	15	15	
<u> </u>	11	11	11	11	11	11	11	11	15	15	
<u></u> <b>£</b> 239.63	11	11	11	11	11	11	11	11	11	11	- 13
ក្ខ 291.02	11	11	11	11	11	11	11	11	11	11	
≥ 353.43	11	11	11	11	11	11	11	11	11	11	
429.23	11	11	11	11	11	11	11	11	11	11	- 12.5
521.28	11	11	11	11	11	11	11	11	11	11	
633.07	11	11	11	11	11	11	11	11	11	11	
768.83	11	11	11	11	11	11	11	11	11	11	- 12
933.7	11	11	11	11	11	11	11	11	11	11	
1133.9	11	11	11	11	11	11	11	11	11	11	
1377.1	11	11	11	11	11	11	11	11	11	11	- 11.5
1672.4	11	11	11	11	11	11	11	11	11	11	
2031.1	11	11	11	11	11	11	11	11	11	11	
2466.6	11	11	11	11	11	11	11	11	11	11	11
	0	1	2	3	4	5	6	7	8	9	
					Imel	-eriod					



📣 MathWorks



## **MathWorks lowers the Reinforcement Learning barrier to entry**

### Learning Resources

📣 MathWorks	Products	Solutions	Academia	Support	Community	Events						
Self-Paced Online Courses												
Home My Courses												

#### Reinforcement Learning Onramp

Take course

Learn the basics of creating intelligent controllers that learn from experience in MATLAB\*. Add a reinforcement learning agent to a Simulink\* model and use MATLAB to train it to choose the best action in a given situation.

Share Course | Share Certificate & Progress | Setting

#### **Reinforcement Learning Onramp**

Wealth Evolution

**Multiperiod GBWM Using** 

**Reinforcement Learning** 



MathWorks® Products Solutions Academia Support Community Events



#### **RL with MATLAB eBook**

Analysis for trainNetwork usage Same: not										
Analysis data: 28-0ct-2021 14 39 03				b.9M tatal learnables	layers warnings error					
	-	ANUTHO RESECT								
e data		Name	Type	Activations	Learnable Properties					
aprof-7x		data 124-124-3 images with "personniar" nor	Image Input	224(5) = 224(5) = 3(C) = 1(8)						
sector.		00m1-7x7_52 547273 considers with strike (2.2) a.	Convolution	112(5) = 112(5) = 64(C) = 1(8)	Meights 7 = 7 = 3 = 64 8585 1 × 1 × 64					
past's br		comit-rate_717	Facu	112(5) = 112(5) = 64(5) + 2(8)						
pedire.	1	positi-3x3_s2 2x3 max pooling with stride 12 2 and pa	Mai Pooling	$56(5) = 58(5) \times 64(C) \times 1(8)$						
aandrik	- 5	podit-nomi cos chamé remaission elle 5 chan	Crass Channel Nor.	56(5) = 56(5) = 64(1) = 1(8)						
• coniQ-m	1	com/2-3x3_metace 04 141404 constitutions with attrate (1.1)	Convolution	56(5) = 56(5) = 64(0) = 1(8)	lacignts 1 = 1 = 64 = 64 fins 1 = 1 = 64					
00%2-la3	7	com/2 relu_3r3_reduce	Facu	56(5) = 58(5) = 64(5) = 1(1)						
conduc.	1	com2-3x3 19212-3x94 constitions with attille (1.1.	Convolution	56(S) = 56(S) = 182(C) = 1(8)	læigtts 3 = 3 = 64 + 182 ties 1 = 1 = 182					
appell-la.	1	com2-mig_3r3	Facu	56(5) = 56(5) = 182(C) = 1(8)						
Pospier. a respire. a respire. a respire.	1	com/2-nam2 com/2-nam2	Crass Channel Nor	$56(5) = 56(5) \times 182(5) \times 1(0)$						
inception, a inception, a inception, a inception,	-	2002-3x3_52 2x3 max posing with stride (2.2) and pa-	Mai Pooling	$20(5) = 20(5) \times 102(5) \times 1(0)$						
irospfon, e irospfon, e irospfon,	12	eception_3a=5x1 Se to t citiz constitutions with strike (1.1.	Convolution	$28(5) \times 28(5) \times 64(0) \times 1(0)$	leights 1 × 1 × 192 × 64 time 1 = 1 = 64					
Pospton, & Pospton	- 1	eception_3e-mix_tx1	Picu	$20(5) \times 20(5) \times 64(1) \times 1(0)$						
- Claim	- 14	Inception_3e-3x3_reduce Int to the CRD constructions with an inter[1,1,1]	Convolution	$28(5) \times 28(5) \times 96(C) \times 1(0)$	Heights 1 × 1 × 192 × 96					

#### Network Analyzer App



Hedging an Option Using Reinforcement Learning Toolbox





#### **Deep Network Designer**



### Deep RL for Optimal Trade Execution

### Low-code Workflows

Examples to get started





