https://www.youtube.com/@AndrejKarpathy, https://www.tensorflow.org/agents/tutorials/0_intro_rl, https://www.tensorflow.org/tutorials/keras/regression, https://www.kdnuggets.com/10-github-repositories-master-reinforcement-learning

Perfect! I now have comprehensive research. Let me create your complete learning pathway with a detailed task breakdown and supplementary resource guide.

Complete RL Factory Scheduling Learning Pathway

Part 1: Sequential Learning Tasks (24 Tasks)

Task #	Title	Learning Objective	Prerequisites	Description	Deliverable	Resource Reference	Estimated Time	Success
1	Python Fundamentals for ML	Establish core Python skills needed for ML projects	None	Write Python scripts demonstrating: variables, loops, functions, list comprehensions, file I/O, and debugging. Start simple (FizzBuzz), progress to data manipulation with dictionaries.	Python script with 5+ working examples	Python Official Docs + Codecademy Python Course	4-6 hours	Can write documen functions understan variable s
2	NumPy Essentials	Master array manipulation and mathematical operations	Task 1	Create NumPy arrays, reshape, slice, perform element-wise operations, use broadcasting, calculate statistics. Build intuition for vectorization.	Jupyter notebook with 10+ NumPy examples	[1] [2] [3] + NumPy Official Docs	3-4 hours	Can resha arrays; us broadcast write vect operation without lo
3	Pandas for Data Handling	Learn to manipulate tabular data for RL observations	Task 2	Load CSV data, filter/select rows and columns, create pivot tables, apply transformations. Practice groupby operations.	Jupyter notebook demonstrating pandas operations on a sample dataset	Pandas Official Docs + DataCamp	3-4 hours	Can read, and trans data effici
4	Matplotlib & Visualization Basics	Create plots for training curves and analysis	Task 2	Plot line graphs, scatter plots, bar charts. Learn to save figures, customize colors/labels, create subplots.	5 different plot types with labels and legends	[4] [5] Matplotlib Docs	2-3 hours	Can creat publicatio quality plo proper lab
5	TensorFlow & Keras Neural Network Fundamentals	Understand deep learning basics for RL agents	Task 1	Build a simple feedforward network (2 hidden layers) using Keras Sequential API. Train on MNIST or similar.	Trained neural network model (`.keras` file) + accuracy report	[6] TensorFlow Regression Tutorial + TensorFlow Core Docs	4-5 hours	Can build compile, a basic ne network

Task #	Title	Learning Objective	Prerequisites	Description	Deliverable	Resource Reference	Estimated Time	Success
6	Backpropagation & Gradient Descent	Understand how neural networks learn	Task 5	Manually implement a simple backpropagation loop using TensorFlow's `GradientTape`. Visualize loss decreasing.	Python script with manual backprop implementation + loss curve plot	[7] [8] [9] [10] + [6]	3-4 hours	Can expla forward/b pass; calc gradients manually
7	Fundamentals of Reinforcement Learning	Learn core RL concepts and terminology	None (Theory)	Study MDPs, states, actions, rewards, episodes, policies. Read about the Bellman equation. Solve a simple MDP by hand.	Written summary (500 words) explaining MDP, Bellman, and value functions	[11] TensorFlow Agents Intro + Sutton & Barto RL Book (Ch 1-3)	5-6 hours	Can defin action, rev explain Be equation
8	Introduction to OpenAI Gym/Gymnasium	Learn the standard RL environment interface	Task 7	Install Gymnasium. Load CartPole-v1, reset it, take 10 random actions, print state/reward/done. Understand the API.	Python script running CartPole with random actions	[12] [13] [14] Gymnasium Docs + TensorFlow Agents Tutorials	2-3 hours	Can creat and step t any built- Gymnasiu environm
9	The RL Loop with CartPole	Execute full RL episode collection	Tasks 7, 8	Run 5 episodes of CartPole with random actions. Record states, rewards, and episode lengths. Plot episode length vs episode number.	Jupyter notebook with episode data and visualization	[11] + [15]	2-3 hours	Can colle analyze R episode d
10	Q-Learning Theory & Intuition	Understand value-based RL methods	Task 7	Study the Bellman optimality equation for Q-functions. Understand $Q^*(s,a)$ and temporal difference updates. Solve a simple gridworld with Q-learning by hand.	Written explanation + hand-solved gridworld solution	[11] DQN Intro + Sutton & Barto (Ch 6)	4-5 hours	Can expla error and updates; s simple MD
11	Implement Tabular Q-Learning	Code a basic Q-learning agent	Tasks 10	Implement Q-learning on FrozenLake-v1 (discrete state/action space). Use ε-greedy exploration. Track rewards over episodes.	Python script with trained Q-table + training curve	[16] DQN Explanation + RL tutorials	3-4 hours	Agent ach >70% suc FrozenLa
12	Deep Q-Networks (DQN) Intuition	Learn how deep learning + Q-learning combine	Task 10	Study experience replay, target networks, and ϵ-greedy exploration. Understand why DQN stabilizes deep RL.	Presentation slides (5-7 slides) explaining DQN architecture	[11] + [16] + [17]	3-4 hours	Can expla experienc target net loss funct
13	Implement DQN from Scratch	Build your first deep RL agent	Tasks 5, 12	Implement DQN on CartPole-v1 using TensorFlow. Include replay buffer, target network, ε-greedy exploration. Train for 500+ episodes.	Trained DQN model + training curve showing convergence	[11] TensorFlow Agents DQN Tutorial + [18]	5-6 hours	Agent sol CartPole (>195 over episodes)

Task #	Title	Learning Objective	Prerequisites	Description	Deliverable	Resource Reference	Estimated Time	Success
14	**[CHECKPOINT]** DQN Experimentation & Hyperparameter Tuning	Learn how hyperparameters affect learning	Task 13	Train DQN with 3 different learning rates (0.001, 0.0001, 0.0005). Compare training curves, final rewards. Document findings.	Comparison plot of 3 training runs + analysis document	Research papers on hyperparameter tuning	4-5 hours	Can ident which hyperpara affect convergen speed
15	Policy Gradient Methods Intuition	Learn policy-based RL fundamentals	Task 7	Study REINFORCE algorithm, policy gradients, and the advantage function. Understand why policy gradient is useful.	Handwritten notes explaining policy gradient from first principles	Sutton & Barto (Ch 13) + [19]	3-4 hours	Can expla gradient t intuitively
16	Actor-Critic Methods (A2C) Concepts	Combine value and policy methods	Tasks 12, 15	Study Actor-Critic architecture: actor (policy), critic (value function), advantage function. Read A2C algorithm.	Diagram of actor-critic architecture with math notation	[20] [19] [21]	3-4 hours	Can expla critic redu variance; actor-criti diagram
17	Train PPO Agent on CartPole	Use a production-ready RL algorithm	Task 13	Train Proximal Policy Optimization (PPO) from Stable Baselines3 on CartPole. Compare to your DQN.	Trained PPO model + comparison plot (PPO vs DQN rewards)	[22] [23] [24] [25] Stable Baselines3 Docs	3-4 hours	PPO achie comparab results tha in fewer s
18	Create Your First Custom Gymnasium Environment	Build a simple custom environment	Tasks 8, 9	Create a "NumberGuessing" environment: agent guesses a number 1-100, receives reward based on closeness. Implement reset, step, render. Register it with Gymnasium.	Custom environment file + Python script training an agent on it	[12] [13] [14] [26] [27] + [28] GitHub examples	4-5 hours	Environm works wit standard Gymnasiu agent lear
19	Design Factory Simulation Environment (Part 1)	Start building your core project environment	Task 18	Create a basic factory environment with 2 functional units (FUs). State: FU availability (busy/free). Actions: route order to FU. Simple reward (completion bonus).	Jupyter notebook with environment code + test runs	[12] [13] [15] Gymnasium Docs + [29] [30] Discrete Event Simulation	5-6 hours	Environm 2 FUs; ag observe s take actio
20	Implement Discrete Event Simulation with SimPy	Add time-based simulation to factory	Task 19	Learn SimPy basics: `env.timeout()`, `yield` generators, processes. Simulate a simple queue system with arrivals and service.	Python script simulating a queue with SimPy	[29] [30] [31] SimPy documentation + tutorials	4-5 hours	Can mode based eve track arrival/ser times

Task #	Title	Learning Objective	Prerequisites	Description	Deliverable	Resource Reference	Estimated Time	Success
21	Integrate RL with Factory Simulation	Combine Gymnasium + SimPy	Tasks 19, 20	Modify your factory environment to use SimPy for time. Order arrives at time T, must complete by deadline. Agent decides order of FU processing.	Factory environment with DES + reward for on-time delivery	[29] [31] [32] FactorySimPy docs	6-7 hours	Environm correctly processin and deadl
22	Multi-Order Scheduling (Dynamic Arrivals)	Extend to your full problem statement	Task 21	Add dynamic order arrivals: Order 1 at t=0, Order 2 at random t. Each has a route (e.g., A→B→D) and deadline. Agent schedules to minimize lateness.	Environment with 2+ orders, dynamic arrivals, Gantt chart visualization	[12] [31] + [33] [34]	7-8 hours	Agent lea schedule orders; vi with Gant
23	Train RL Agent on Factory Problem	Benchmark your first solution	Task 22	Train DQN/PPO on your factory environment. Run 1000+ episodes. Track cumulative reward, on-time delivery rate, FU utilization.	Trained model + metrics dashboard (reward curve, utilization)	[24] [35] [25] Stable Baselines3 examples	6-7 hours	Agent ach >60% on- delivery; learning
24	**[PROJECT CHECKPOINT]** Visualization & Poster Preparation	Create outputs for academic presentation	Task 23	Generate: (1) Gantt chart for a test episode, (2) Training curves (reward + loss), (3) Utilization breakdown by FU, (4) Comparison table: baseline vs RL agent.	High-quality plots suitable for poster; summary document	[4] [5] [36] Gantt chart tutorials + Matplotlib	4-5 hours	Plots are publicatio quality; te story

Part 2: Advanced Extensions (8+ Optional Tasks)

Task #	Title	Description	Prerequisite	Decision Point
25	**[DECISION]** Algorithm Selection: DQN vs PPO vs A2C	Compare 3 algorithms on your factory problem. Which learns fastest? Which is most sample-efficient?	Task 23	Choose one for deep dive
26	Reward Shaping for Factory Domain	Design shaped rewards: bonus for early delivery, penalty for machine idle. Measure impact on learning.	Task 23	**Stretch**: Optional refinement
27	Multi-Objective Rewards	Implement weighted rewards: (1) on-time delivery, (2) machine utilization, (3) throughput. Study trade-offs.	Task 24	Enables more realistic scheduling
28	Transfer Learning: Generalize to Larger Factories	Train on 4 FUs, test on 6 FUs without retraining. Or train on one order pattern, test on new patterns.	Task 23	**Stretch**: Research direction
29	Multi-Agent RL (Optional)	Train separate agents for each FU (decentralized) vs centralized scheduler. Compare coordination.	Task 22	**Decision**: Single vs multi-agent
30	Reliability & Degradation	Add machine failures (random downtime). Agent must reschedule. Measure robustness.	Task 22	Aligns with your planned extensions
31	Buffer/Queue Modeling	Add intermediate buffers between FUs. Agent manages queue lengths. Observe congestion handling.	Task 21	Advanced constraint
32	Benchmark Against Classical Methods	Compare RL agent to: (1) greedy dispatching rules, (2) constraint programming (OR-Tools), (3) genetic algorithms.	Task 24	Validates RL approach

Part 3: Supplementary Resource Guide

Skill/Topic	Resource Type	Resource Name & Link	Relevant Tasks	Why This Resource	Priority
Python Fundamentals	Online Course	Python.org Official Tutorial	Tasks 1, 2	Authoritative, free, comprehensive	**Essential**
Python Fundamentals	Tutorial	Codecademy Python Course	Task 1	Interactive, beginner-friendly	**Essential**
NumPy	Documentation	NumPy Official Docs (numpy.org)	Task 2	Complete reference with examples	**Essential**
NumPy	Video Tutorial	NumPy Tutorial by Keith Galli (YouTube)	Task 2	Visual, walkthrough-based learning	**Recommended**
Data Manipulation	Course	DataCamp: Data Manipulation with Pandas	Task 3	Practical pandas focus	**Recommended**
Visualization	Tutorial	DataCamp: Gantt Charts with Matplotlib	[4] Tasks 4, 24	Specific to your project needs	**Essential**
Visualization	Documentation	Matplotlib Official Docs	Task 4	Complete reference	**Recommended**
Deep Learning Basics	Tutorial	[6] TensorFlow Regression Tutorial	Task 5	Official TensorFlow tutorial, hands-on	**Essential**
Deep Learning	Course	Coursera: Deep Learning Specialization (Andrew Ng)	Task 5	Comprehensive DL fundamentals	**Recommended**
Neural Networks	Video Series	[7] 3Blue1Brown Neural Networks Playlist (YouTube)	Task 5, 6	Exceptional visual explanations	**Recommended**
Backpropagation	Article	[9] Matt Mazur: Step-by-Step Backpropagation Example	Task 6	Detailed worked example	**Essential**
Backpropagation	Google Course	[10] Google ML Crash Course: Backpropagation	Task 6	Official, clear explanation	**Recommended**
RL Theory	Textbook	[11] TensorFlow Agents: Intro to RL & DQN	Tasks 7, 10, 12	Official, directly applicable	**Essential**
RL Theory	Textbook	Sutton & Barto: Reinforcement Learning (2nd Edition)	Tasks 7, 10, 15	Bible of RL, comprehensive	**Essential**
RL Theory	Video Series	[11] Andrej Karpathy YouTube Channel (Talks on ML/AI)	Task 7	Expert perspective, deep dives	**Recommended**
Gymnasium/Gym	Documentation	[14] Gymnasium Official Docs	Tasks 8, 18	Complete API reference	**Essential**
Gymnasium	Tutorial	[13] YouTube: Custom Gymnasium Environment Tutorial	Task 18	Step-by-step video walkthrough	**Recommended**
Gymnasium	GitHub	[28] johnnycode8/gym_custom_env Repository	Task 18	Multiple working examples	**Recommended**
Q-Learning	Article	[16] Lukasschwarz.de: DQN Explanation & Implementation	Tasks 10, 12, 13	Detailed math + TensorFlow code	**Essential**
DQN	Paper	Mnih et al. 2015: Playing Atari with Deep Reinforcement Learning	Task 12	Original DQN paper	**Optional**
Stable Baselines3	Documentation	[22] [23] [24] [25] Stable Baselines3 Docs	Tasks 13, 17	Complete API + examples	**Essential**
Stable Baselines3	Tutorial	[35] SB3 Examples Documentation	Task 17	Ready-to-run code examples	**Recommended**
Policy Gradients	Article	[19] Hugging Face: Advantage Actor Critic (A2C)	Tasks 15, 16	Clear visual explanations	**Essential**
Actor-Critic	Article	[20] CodeSignal/APXML: A2C Explained	Task 16	Mathematical + intuitive explanation	**Recommended**
Actor-Critic	GeeksforGeeks	[21] GeeksforGeeks: Actor-Critic Algorithm	Task 16	Tutorial with pseudocode	**Recommended**

Skill/Topic	Resource Type	Resource Name & Link	Relevant Tasks	Why This Resource	Priority
Custom Environments	Tutorial	[26] CTO2B: OpenAI Gym Custom Environments Guide	Task 18	Step-by-step guide	**Essential**
Discrete Event Simulation	Video	[29] Paul Fishwick: Factory DES in Python (YouTube)	Task 20	Classic factory simulation example	**Essential**
Discrete Event Simulation	Library Docs	[31] SimPy Official Documentation	Task 20	Complete reference	**Essential**
DES Tutorial	DataCamp	[30] DataCamp: DES with SimPy	Task 20	Practical walkthrough	**Recommended**
DES for Manufacturing	Library	[32] FactorySimPy Documentation	Task 21	Purpose-built for manufacturing	**Recommended**
Manufacturing Scheduling	Paper	[33] Deep RL for Semiconductor Scheduling (IEEE)	Task 22	Real manufacturing case study	**Recommended**
Manufacturing Scheduling	Paper	[34] RL for Real-Time Scheduling (Taghipour et al.)	Tasks 22, 32	DQN + MBPO on actual factory data	**Recommended**
Job Shop Scheduling	Documentation	[37] OR-Tools: Job Shop Scheduling	Task 32	Constraint programming baseline	**Recommended**
Job Shop Scheduling	Article	[38] AMPL: Job Shop Scheduling Intro	Task 32	Operations research perspective	**Recommended**
Reward Shaping	Article	[39] CodeSignal: Reward Shaping for Faster Learning	Task 26	Practical reward design principles	**Recommended**
Reward Shaping	Paper	[40] ArXiv: Comprehensive Overview of Reward Engineering	Tasks 26, 27	Detailed taxonomy of techniques	**Optional**
Multi-Agent RL	Course	[41] Hugging Face: Intro to Multi-Agent RL	Task 29	Free, visual course	**Recommended**
Multi-Agent RL	Wikipedia	[42] Multi-Agent Reinforcement Learning Overview	Task 29	Broad conceptual overview	**Recommended**
Multi-Agent RL	GeeksforGeeks	[43] Multi-Agent RL in AI	Task 29	Tutorial with examples	**Recommended**
Multi-Agent RL	Paper	[44] Survey: Cooperative MARL in Open Environments	Task 29	State-of-the-art review	**Optional**
Version Control	Documentation	GitHub: Git Basics	All tasks	Essential for saving work	**Essential**
Academic Writing	Tool	Overleaf: LaTeX for Papers	Tasks 24, 32	Documentation in LaTeX	**Recommended**
Code Notebooks	Tool	Google Colab (already using)	All tasks	Recommended for running code	**Essential**

Part 4: Task Flow & Decision Points

Linear Path (Minimal, 15 tasks)

Tasks 1 → 2 → 7 → 8 → 9 → 10 → 13 → 18 → 19 → 21 → 22 → 23 → 24

Comprehensive Path (24 tasks + checkpoints)

Follow all tasks 1-24 in order. Checkpoints at Tasks 14 & 24 allow you to decide whether to extend further.

Key Decision Points

After Task 14 (DQN Tuning):

- ✅ Proceed to policy gradients if you want to explore diverse algorithms
- ▯ Skip to Task 18 if you want to focus on custom environments

After Task 17 (PPO Training):

- **Choose Task 25 Algorithm Comparison** if you want rigorous benchmarking
- **Skip directly to Task 18** if you want to move to your custom domain

After Task 23 (First Factory Agent):

- **Mandatory: Task 24 (Checkpoint & Visualization)** — needed for poster
- **Optional: Task 26-32** — advanced extensions based on research goals

Part 5: Poster-Ready Outputs

Tasks that generate **publication-quality** deliverables:

- **Task 14:** Hyperparameter comparison chart
- **Task 24:** Gantt chart visualization + training curves
- **Task 27:** Multi-objective trade-off plot
- **Task 32:** RL vs classical baseline benchmark table

Part 6: Methodology Documentation Guide

For your academic write-up, document:

1. **After Task 7:** "Reinforcement Learning Fundamentals Acquired"
2. **After Task 13:** "Deep Q-Networks Implementation & Tuning Process"
3. **After Task 22:** "Factory Scheduling Environment Design"
4. **After Task 23:** "Agent Training & Performance Analysis"
5. **After Task 24:** "Results, Visualizations, & Preliminary Findings"

Quick Start Recommendation

Week 1: Tasks 1-7 (Python + RL Theory)
Week 2: Tasks 8-13 (Gymnasium + DQN)
Week 3: Tasks 18-21 (Custom Environments + SimPy)
Week 4: Tasks 22-24 (Factory Problem + Poster)

This gives you a **working RL agent on your factory problem in 4 weeks**, with time to extend further if desired.

Would you like me to:

1. **Elaborate on any specific task** with code snippets or detailed instructions?
2. **Create a companion Jupyter notebook** for Tasks 1-7?
3. **Provide a template structure** for your factory environment code?
4. **Generate additional resources** on any particular skill area?

Good luck with your project!

⁂

1. https://www.tutorialspoint.com/numpy/numpy_array_manipulation.htm
2. https://www.programiz.com/python-programming/numpy/array-functions
3. https://numpy.org/doc/stable/reference/routines.array-manipulation.html
4. https://www.datacamp.com/tutorial/how-to-make-gantt-chart-in-python-matplotlib
5. https://www.datacamp.com/de/tutorial/how-to-make-gantt-chart-in-python-matplotlib
6. https://www.tensorflow.org/tutorials/keras/regression
7. https://www.geeksforgeeks.org/deep-learning/back-propagation-with-tensorflow/
8. https://www.datacamp.com/tutorial/mastering-backpropagation
9. https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/
10. https://developers.google.com/machine-learning/crash-course/neural-networks/backpropagation

11. https://www.tensorflow.org/agents/tutorials/0_intro_rl
12. https://gymnasium.farama.org/introduction/create_custom_env/
13. https://www.youtube.com/watch?v=AoGRjPt-vms
14. https://gymnasium.farama.org/tutorials/gymnasium_basics/environment_creation/
15. https://www.datacamp.com/tutorial/reinforcement-learning-with-gymnasium
16. https://lukasschwarz.de/dqn
17. https://github.com/tensorflow/agents/blob/master/tf_agents/agents/dqn/dqn_agent.py
18. https://www.tensorflow.org/agents/tutorials/1_dqn_tutorial
19. https://huggingface.co/blog/deep-rl-a2c
20. https://apxml.com/courses/intermediate-reinforcement-learning/chapter-5-actor-critic-methods/advantage-actor-critic-a2c
21. https://www.geeksforgeeks.org/machine-learning/actor-critic-algorithm-in-reinforcement-learning/
22. https://stable-baselines3.readthedocs.io/en/master/modules/dqn.html
23. https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html
24. https://stable-baselines3.readthedocs.io
25. https://github.com/DLR-RM/stable-baselines3
26. https://cto2b.io/blog/custom-openai-gym-environments/
27. https://vadim.me/publications/heartpole/
28. https://github.com/johnnycode8/gym_custom_env
29. https://www.youtube.com/watch?v=G2WftFiBRFg
30. https://campus.datacamp.com/courses/discrete-event-simulation-in-python/developing-discrete-event-models-using-simpy?ex=3
31. https://towardsdatascience.com/object-oriented-discrete-event-simulation-with-simpy-53ad82f5f6e2/
32. https://factorysimpy.github.io/FactorySimPy/
33. https://www.sciencedirect.com/science/article/abs/pii/S0957417421015359
34. https://journals.sagepub.com/doi/full/10.1177/16878132221086120
35. https://stable-baselines3.readthedocs.io/en/master/guide/examples.html
36. https://coderzcolumn.com/tutorials/data-science/gantt-chart-using-matplotlib
37. https://developers.google.com/optimization/scheduling/job_shop
38. https://ampl.com/mo-book/notebooks/03/job-shop-scheduling.html
39. https://codesignal.com/learn/courses/advanced-rl-techniques-optimization-and-beyond/lessons/reward-shaping-for-faster-learning-in-reinforcement-learning
40. https://arxiv.org/html/2408.10215v1
41. https://huggingface.co/learn/deep-rl-course/en/unit7/introduction-to-marl
42. https://en.wikipedia.org/wiki/Multi-agent_reinforcement_learning
43. https://www.geeksforgeeks.org/machine-learning/multi-agent-reinforcement-learning-in-ai/
44. https://arxiv.org/abs/2312.01058
45. https://www.youtube.com/@AndrejKarpathy
46. https://www.digitalocean.com/community/tutorials/creating-custom-environments-openai-gym
47. https://www.gymlibrary.dev/content/environment_creation/
48. https://pypi.org/project/simprocesd/
49. https://www.tensorflow.org/agents/tutorials/1_dqn_tutorial?hl=ja
50. https://notebook.community/tensorflow/agents/docs/tutorials/1_dqn_tutorial
51. https://www.youtube.com/watch?v=bD6V3rcr_54
52. https://github.com/DLR-RM/stable-baselines3/blob/master/docs/modules/ppo.rst
53. https://www.geeksforgeeks.org/machine-learning/optimizing-production-scheduling-with-reinforcement-learning/
54. https://stable-baselines3.readthedocs.io/en/master/guide/rl_tips.html
55. https://pettingzoo.farama.org/tutorials/sb3/index.html
56. https://stackoverflow.com/questions/60192147/how-to-get-a-gannt-chart-using-matplotlib
57. https://www.youtube.com/watch?v=RKSMzIUcJMA
58. https://stable-baselines3.readthedocs.io/_/downloads/en/master/pdf/
59. https://www.geeksforgeeks.org/python/python-basic-gantt-chart-using-matplotlib/
60. https://www.sciencedirect.com/science/article/pii/S027861252400253X

61. https://stable-baselines3.readthedocs.io/en/master/guide/custom_policy.html
62. https://www.youtube.com/watch?v=jTS-nGNYb20
63. https://www.machinelearningmastery.com/tutorial-first-neural-network-python-keras/
64. https://www.atmosera.com/blog/building-neural-networks-with-keras-and-tensorflow/
65. https://www.youtube.com/watch?v=8gFrim7cTzI
66. https://www.geeksforgeeks.org/python/python-operations-on-numpy-arrays/
67. https://proceedings.mlr.press/v139/wen21b/wen21b.pdf
68. https://stackoverflow.com/questions/47416861/backward-propagation-in-keras
69. https://numpy.org/doc/2.2/reference/routines.array-manipulation.html
70. https://warwick.ac.uk/fac/sci/dcs/people/long_tran-thanh/online-lectures/4-policy_gradient.pdf
71. https://www.geeksforgeeks.org/machine-learning/backpropagation-in-neural-network/
72. https://www.reddit.com/r/reinforcementlearning/comments/15t28y5/multiagent_reinforcement_learning/
73. https://github.com/tshrjn/env-zoo
74. https://adasci.org/all-you-need-to-know-about-multi-agent-reinforcement-learning/
75. https://www.youtube.com/watch?v=UGvc-qujB-o
76. https://arxiv.org/abs/2011.02669
77. https://arxiv.org/abs/2312.10256
78. https://www.reinforcementlearningpath.com/reward-shaping/
79. https://arxiv.org/html/2407.18111v1
80. https://www.youtube.com/watch?v=qgb0gyrpiGk
81. https://gibberblot.github.io/rl-notes/single-agent/reward-shaping.html
82. https://en.wikipedia.org/wiki/Job-shop_scheduling
83. https://www.marl-book.com