Evmbodied AI:
A Survey of Embodied AI: From Simulators to Research Tasks
[https://arxiv.org/abs/2103.04918]
=>Here we can target one of the tasks mentioned in the above review.

=> RL is in general more mathematically rigorous than ML/DL, So see if you can get through some of these. If you plan to come up with your own algorithm/model some time down the lane you shouldn't be scared of math.

Reinforcement Learning Courses:
https://www.davidsilver.uk/teaching/ by David Silver (For understanding what all RL is about)
https://rail.eecs.berkeley.edu/deeprlcourse/ by Sergey Levine (Advanced RL - Based on Deep Learning)

RL in General:

1. Trust region policy optimization: deep RL with natural policy gradient and adaptive step size (TRPO)
2. Proximal policy optimization algorithms: deep RL with importance sampled policy gradient (PPO)
3. Dueling network architectures for deep reinforcement learning: separates value and advantage estimation in Q-function. (DDQN)
4. Sim-to-Real: Learning Agile Locomotion For Quadruped Robots.

Model Based RL

1. Guided Policy Search (https://graphics.stanford.edu/projects/gpspaper/gps_full.pdf)
2. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models (https://arxiv.org/abs/1805.12114)

=> Try to understand why we need model-based RL or Policy Gradients..etc.

Task-1 (14-12-2024):

1. Playing Atari with Deep Reinforcement Learning (https://arxiv.org/pdf/1312.5602)
    a. Try reproducing the graphs given in the paper.
    b. Try to follow the OPEN AI GYM framework. Start with a Cart-Pole example to develop understanding. Then go with a PONG game.
    c. Ponder on what all weaknesses of the method and possible extensions. Two well cited extensions are
        i. H. van Hasselt, A. Guez, and D. Silver, *"Deep Reinforcement Learning with Double Q-learning"* (2016).
        ii. Z. Wang et al., *"Dueling Network Architectures for Deep Reinforcement Learning"* (2016).

d. If possible, integrate Double DQN into your work and compare results. The integration is simple yet powerful. Then, you can try Dueling DQN

Date : 23/11/2024

Disaster zone (AV navigation)

High level overview on [https://arxiv.org/abs/2103.04918] :

- **Simulator repositories :**

1. DeepMind Lab : https://github.com/google-deepmind/lab
2. AI2-THOR : https://github.com/allenai/ai2thor
3. CHALET : https://github.com/lil-lab/chalet
4. VirtualHome : https://github.com/xavierpuigf/virtualhome
5. HabitatSim : https://github.com/facebookresearch/habitat-sim
6. iGibson : https://github.com/facebookresearch/habitat-sim
7. Sapien : https://github.com/haosulab/SAPIEN
8. ThreeDWorld : https://github.com/threedworld-mit/tdw
9. VR kitchen : https://github.com/xfgao/VRKitchen

- **Major downstream tasks in Embodied AI :**

1. Visual Exploration

   Case 1 : Exploration before with limited steps before navigation.
   Case 2 : Agent builds the map concurrently as it navigates the space.

   Approaches :

   1. Curiosity : Agent seeks states that are difficult to predict and prediction error is used as a reward signal.

2. Coverage : Maximize the amount of targets directly observed (SLAM and Transformer self-attention techniques).
3. Reconstruction : Agent recreates other views from an observed view

Popular datasets :

1. https://github.com/niessner/Matterport
2. https://svl.stanford.edu/igibson/

2. Visual Navigation

An agent navigates a 3D environment to a goal with or without external priors or natural language instruction.

Classical navigation involves localization, path planning and locomotion. Whereas, embodied AI aims to learn these navigation systems from data, so as to reduce case-specific downstream tasks.

1. Point Navigation : An agent is initialized at the origin (0,0,0) in an environment, and the fixed goal point is specified (x,y,z).