Module 8

Symbolic Planning

▼ What is symbolic planning?

- It uses physical symbols and discrete, pre-defined logical rules to perform a task the undertaking of which has been pre-planned.
- The process of using symbolic methods to develop planned procedures based on rulesets and symbolic knowledge representation in order to fulfill a task in a real-world (i.e., grounded) environment.

▼ What is reinforcement learning (RL) a form of?

a form of unsupervised learning, which just means data is unlabeled

▼ How is reinforcement learning essentially work?

- A reward system is set up, allowing the developer to define which results should be "rewarded" with a reward value and how much.
- This allows the model to define its own path to solving the problem, by seeking actions that result in rewards.

▼ How does self-supervised learning essentially work?

- Like unsupervised learning, self-supervised learning uses data that is initially unlabelled.
- However, unlike unsupervised learning, the data does not remain unlabelled, instead, the model labels the data itself based on predefined parameters, input from sensor modalities, or relations between samples.

▼ What is the physical symbol system hypothesis (PSSH)?

a physical symbol system has the necessary and sufficient means for general intelligent action

▼ What is included in a physical symbol system?

Classes or categories of objects, rulesets, and logic formulations

▼ What is the symbol grounding problem?

The essence of the argument is that symbols must have some meaning in a real environment, rather than just being abstract concoctions existing only in relation to other similarly ungrounded symbols.

▼ An example of symbolic planning systems?

- robotic control
- e.g. route planning
  - What are the constraints?
  - Is the robot to find the shortest path? Is it more important to take the shortest path, or is it more important to not bump into things?
  - How are these decisions prioritized?
  - What if the unexpected happens or the environment changes?

▼ What do action languages offer?

they allow the developer to specify state transitions, build a formal model of the task environment, and define the rules, e.g. a robot's behavior

▼ What are some weaknesses of reinforcement learning and how can they be solved by Hierarchical Reinforcement Learning (HRL)?

- Scaling up:
  - When a problem is large, it is not as simple as adding more parameters. At a certain point, ANNs become difficult to manage as single models.
  - By placing HRL sub-models together in a planned format, scaling up can be achieved more readily with HRL.
- Abstraction:

    - Complex problems require abstraction using "smaller" concepts of finer detail.
    - By using many specialized ANNs, HRL can solve this problem by solving each sub-task individually before passing the task on to the appropriate next network in line.
- Generalization:
    - RL tends to be overspecialized for complex tasks, which can hamper transfer learning and the ability to adapt to new environments after the initial training.
    - By using HRL, you can subdivide these tasks into smaller tasks, which can be transferred more easily.