Modular Arm Puzzle
Info Sheet

***Materials Needed**: Modular Arm Puzzle Mat and Shapes.pdf (**printed at full scale**), pencil with eraser, scissors, optional: colored pencils or crayons.*

Activity Overview

1. Review key terms and background information:

    Machine Learning: a form of artificial intelligence in which a program finds patterns in data, and applies those patterns to guide its future decisions.

    Reinforcement Learning: a way of creating the data to learn from by informed trial and error.

    Modular robots can allow a small set of components to be recombined into many different designs. One challenge, then, is picking the best design for each environment or job.

    A common robot arm job is to pick up an object in one location and place it in another. There might be obstacles in the way, and we have a set of modules to work with. Here, we will play the role of the computer, which is to learn how to construct a robot for each new job.

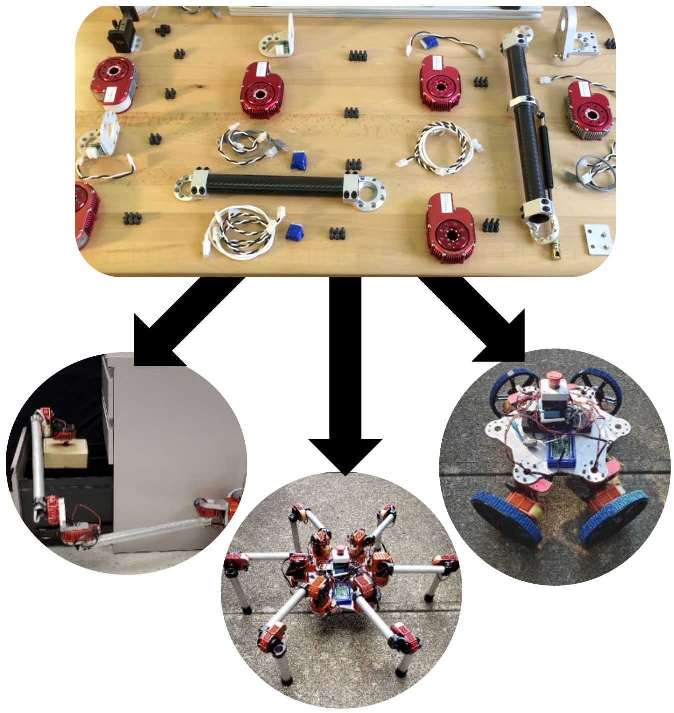
    Reachable Workspace: The set of points that a single arm design can reach.

2. Print out the puzzle mat and pieces. Make sure to print at full scale!
3. Color module parts if desired, then cut out the modules along the outer edges.
4. Go over directions for first part of the activity. A start, goals, and obstacles are specified, and the objective is to come up with a robot (i.e. a connected series of modules) that can reach the goals without colliding with an obstacle.
5. Aid students in working through the puzzle and explore the thought process that goes through solving it.
6. Go over directions for the second part of the activity. Students will create their own problem and solution. They will then offer this problem to another student to solve.
7. Repeat as time allows. The number of goal points can be increased slowly to increase the difficulty.
8. Complete the questions and discuss the results.

Carnegie Mellon University
Leonard Gelfand Center +
College of Engineering

Modular Arm Puzzle Background

Modular robots are made of interchangeable parts. They allow you to make many different robots from a small set of parts. This is kind of like LEGOs, but not only do they connect physically, they must also connect electronics and programming. Some research also has the robot "self-reconfigure" like transformers, but those are mostly still experimental.



Images: top, modules from Hebi Robotics, and bottom, robots with arms, legs, and wheels constructed and photographed by Julian Whitman

Carnegie Mellon University
Leonard Gelfand Center +
College of Engineering

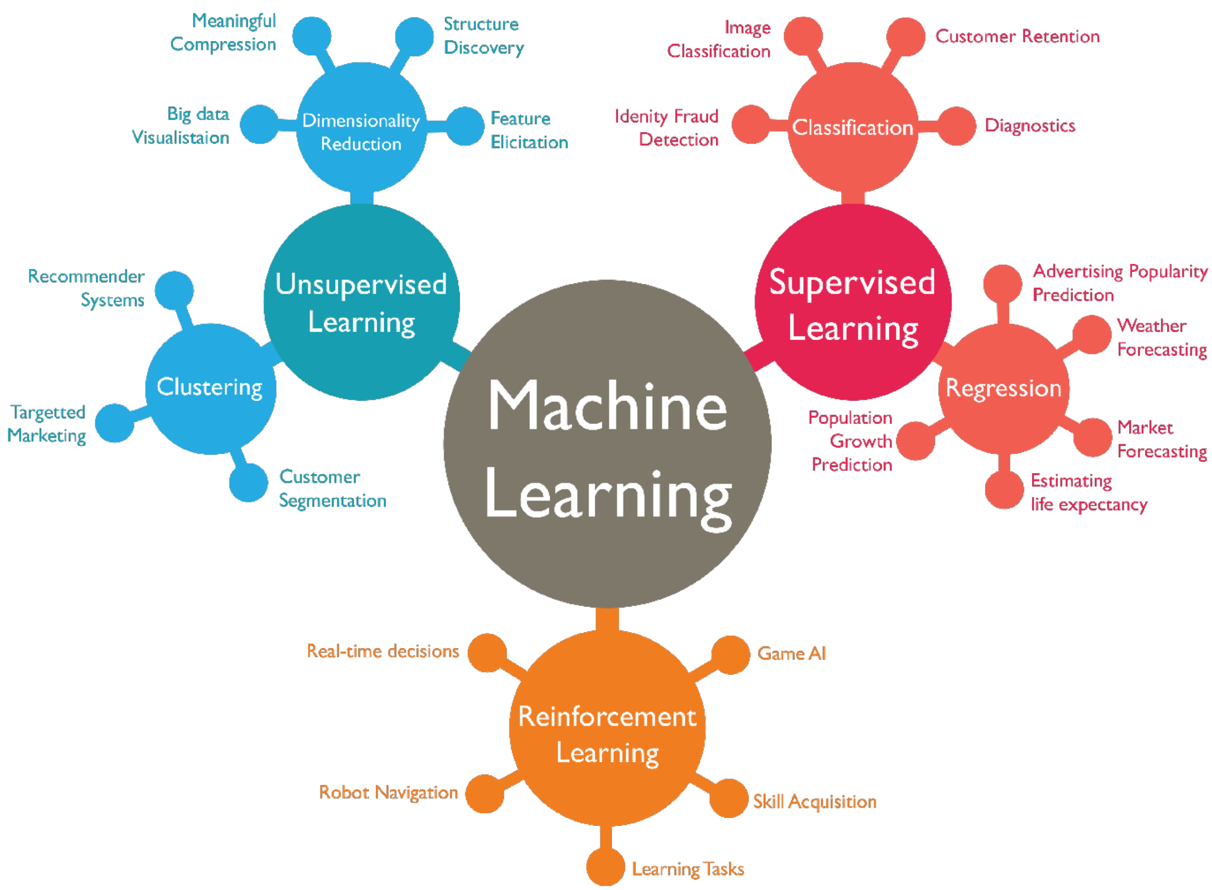
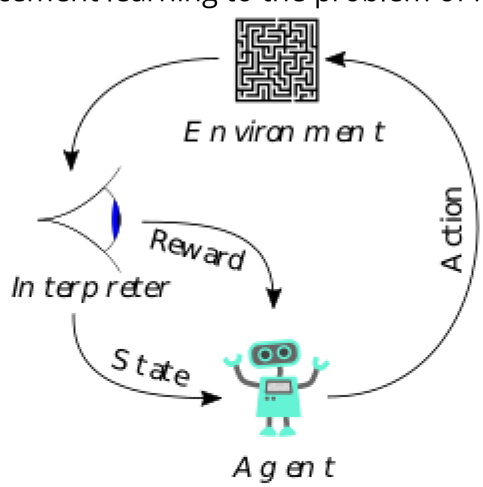


Image credit: https://www.slideshare.net/awahid/big-data-and-machine-learning-for-businesses

There are a few types of Machine learning algorithms, which differ based on the type of data and signals available to the algorithm. These lead to different applications. Julian researches applying reinforcement learning to the problem of robot design.



A cartoon high-level view or reinforcement learning. Image credit:
[*https://en.wikipedia.org/wiki/Reinforcement_learning*](https://en.wikipedia.org/wiki/Reinforcement_learning)

Carnegie Mellon University
Leonard Gelfand Center +
College of Engineering

> Wikipedia defines: “Reinforcement learning (RL) is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward. Reinforcement learning is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning. Reinforcement learning differs from supervised learning in not needing labelled input/output pairs be presented, and in not needing sub-optimal actions to be explicitly corrected. Instead the focus is on finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge).”

In our case, the “reward” that we want to maximize is the ability for the arm to reach the target goals without collisions. We could also add additional objectives, for instance, reaching the goals with the fewest modules.

Modular Arm Puzzle Activity

In the following activity, we play the role of the robot designer. Our goal is to select the modules (components and order) and the control (the joint angles between the modules) to reach a set of points.

Game Rules

- No part of the modular arm can overlap with the obstacles (Areas shaded with black squares to form the letters “C, M, & U”)
- Modular arm parts must connect at the circular “joints”, but are otherwise unable to overlap. Only one connection per joint is allowed.
- The first joint on the first module must connect to the starting point, the filled circle in the center.
- The last joint on the last module must connect to the goal.
- Each module, obstacle, and marker must be correctly aligned with the grid.
- All pieces must be fully on the board, and aligned with the grid.
- To solve a goal point, the final joint (circle at the last space) must be located at that point. To solve the set, the same modular arm must reach all of the points listed.
- The sequence of modules used within a single set of goal points must be the same, but the orientation (representing the angle of the joints) of the joints can vary.

Part 1 Objective: Solve the Board

For the following puzzles, place the start marker, goal markers, and obstacles as listed. Then, attempt to create an arm, following the rules above, that can reach from the start to the goals.

The puzzle examples, repeated on the activity sheet, are:

Task 1: C8
Task 2: C15, D16
Task 3: I4, G4, I12

Marking a Goal: For each goal point in the set, use a pencil to lightly draw an "X" on the spots to mark that the arm should reach. Then, when you are finished with that set, erase those marks with the pencil.

Tips:

1. Talk through your process in these, but don't give away the answer to your neighbor's until everyone has either gotten the answer or given up.
2. Note that if the pieces are not printed and cut correctly, it may not be possible to complete the puzzles.
3. There may be multiple solutions to the puzzle as well. If so, could some solutions be better than others?

Tie-breakers could be:

- Whichever design uses the fewest modules (number of parts)
- Whichever design uses the lowest total robot volume (measured in grid squares occupied)
- Whichever design has the most similar joint angle difference between goal points (measured in number of joint turns to move from between the goal points).

These are all relevant for real robots as well!

4. Prompt students to consider what makes the problem hard, and how they learn to solve it by multiple attempts. What factors to they consider explicitly, and what factors are guided by spatial intuition?

Part 2: Solve Another Person's Puzzle

Using the puzzle board components, create your own puzzle, and record it in the same syntax as above. At the same time, record the solution to this puzzle.

Puzzle Solving Strategy:

1. Note that a puzzle is not considered valid unless it comes with at least one valid solution!

2. Think about it first! Don't list random points, because that may not yield a solution!

3. One way to ensure that you are making a valid puzzle is by using the "forward" problem:
    a. Take a combination of parts and see where it can reach.
    b. Then, pass the "inverse" problem to your neighbor, and see if they can solve it!

    (Note: this can be a strategy used in teaching the computer to solve the puzzle with RL, by generating sets of random points using random designs, and which provides solutions to the inverse problem for the points reached by those random designs)

If time allows, conduct multiple rounds, where each one gets harder, for instance:

- On the first round, use at most two goal points and arms with at most three parts.
- On the subsequent rounds, allow additional goals and parts.

NOTE: Some students may be better at this type of spatial puzzle, and the problem can be difficult to solve by hand, so it is ok if there is a range of puzzles that are solved or not solved.

Conclusions

- The "training" step in the neural networks could involve randomly generating and attempting to solve these puzzles.
- It is difficult to describe spatial intuition, but are there any descriptions that get in the right direction? If they sound reasonable, most likely these could be used as search heuristics either the help train the network (so you are not learning from scratch) and these might also be methods already used in the literature.