**Graded Discussion 9**

Reinforcement learning models are trained to maximize a reward function. The model uses experience to learn policies which have increased reward function values, known as improved policies. However, for many real-world problems, creating a good reward function can be difficult. These difficulties can include defining a single task the RL agent can perform, rewarding only the desired behavior and not encouraging spurious behavior, and maintaining safety when encountering unexpected state or incomplete state information.

Think of an RL application you might like to create. What reward function could you use to ensure only desired behavior? How can prevent unexpected behavior, including unsafe behavior, when the RL agent is presented with unusual or incomplete state information?

------------------------------------------------------------------

I do not have any experience working with Reinforcement Learning, but it is clear that writing a reward function is a difficult task especially as not only has to be optimized for the system to learn the most optimal way, but also ensure safety is maintained and any spurious behavior is kept to a minimum. While reading about this topic I encountered an effect that is well known by RL practitioners, which is call the cobra effect or the perverse incentive effect which basically is an incentive that unintentionally rewards people/systems for making the issue worse, the example given is that the British government tried to incentivize people to assist them in getting rid of cobras in some areas of India. If the citizens brought in a venomous snake, they had killed they would be rewarded. After some time, people started breeding these venomous snakes to obtain the reward. These effects are seeing in RL all the time when unforeseen events start to appear as the system starts exploring the environment with the given constraints and rewards. Some of the key points that I found thru the multiply examples that I saw where: 1) Think about the extreme cases first or the possible unsafe situations that the system might encounter, such as a car looking for the fastest route and getting to rewarded for jumping off a cliff, 2) also include these safety events as part of the reward function, 3) Test the RL model with novel situations to see the reaction and look for possible pitfalls. Something to keep in mind based on common good practices for RL is that you get what you incentivize, not what you intend.

**Reference:**

Reinforcement Learning Series: <https://www.youtube.com/watch?v=nyjbcRQ-uQ8&list=PLZbbT5o_s2xoWNVdDudn51XM8lOuZ_Njv>

Reinforcement Learning - Developing Intelligent Agents: <https://deeplizard.com/learn/video/nyjbcRQ-uQ8>

How to Design a Reinforcement Learning Reward Function for a Lunar Lander: <https://towardsdatascience.com/how-to-design-reinforcement-learning-reward-function-for-a-lunar-lander-562a24c393f6>

Deep Reinforcement Learning Models: Tips & Tricks for Writing Reward Functions: <https://medium.com/@BonsaiAI/deep-reinforcement-learning-models-tips-tricks-for-writing-reward-functions-a84fe525e8e0>

Perverse incentive behavior: <https://en.wikipedia.org/wiki/Perverse_incentive#The_original_cobra_effect>

How to make a reward function in reinforcement learning? <https://stats.stackexchange.com/questions/189067/how-to-make-a-reward-function-in-reinforcement-learning>

Writing successful reward functions: <https://nealanalytics.com/blog/writing-successful-reward-functions/>

Deep Learning and Reward Design for Reinforcement Learning: <https://deepblue.lib.umich.edu/bitstream/handle/2027.42/136931/guoxiao_1.pdf>