Assignment 2

Your task, put simply, is to design a reinforcement learning algorithm to teach the mouse how to find the food.

The fundamental task is as follows:

- There is a 100x100 matrix representing a grid where any space can either be occupied by a mouse or a piece of food. There is only one mouse, and an arbitrary amount of food.
- The mouse is able to sense the food with a 3x3 matrix representing its sense of smell, the range of which is the full grid, and stacks (Two foods next to each other will generate twice as much smell). This will be the input for your algorithm. Note that the center of this matrix will always be 0, as that space represents the mouse itself.
- The mouse has a limited amount of energy, which is fully replenished when it finds food. If it runs out of energy, it dies, and the game is over.
- The mouse is able to move in any cardinal direction (North, South, East, and West). The goal is for them to eat all of the food in the grid as quickly as possible.

This simulation is visually represented using PyGame.

This task can also be solved by a trivial algorithm using nothing but simple arithmetic, as the 'scent' of food is a function of distance from the mouse. You can try to find a non RL solution and compare it to the best version of the RL algorithm's results.

For each frame, a forward pass is run through your model.

The input for each frame is an array (or tuple) with four numbers between 0 and 1. These are the probabilities generated from your model. These will determine how the mouse moves.

Brainstorming session 1

1: Does the order matter for the reinforcement learning model? (Ex: Inputting (N,S,E,W) vs (N,E,S,W))

2: Does the order matter for a closed form solution?

3: What would be better for reinforcement learning; taking the highest value from the array as the movement choice, or choosing a random direction weighted by the given probabilities? Why?

Tasks for the students

- **Write a reward function.**

You have access to the current game state, as well as the number of previous frames (Ex: 50) of input matrices, food level, and number of food tiles found. This number of frames will be the same number of frames it takes to starve from a full energy level.

This can be very simple, like just using the number of food tiles found, or use reward shaping using multiple frames of previous input matrices to create more frequent positive/negative rewards (Ex: when the mouse moves closer or farther away from food.)

Brainstorming session 2:

Q: What would be a sparse reward function for this model?

Q: How can the reward function be improved?

- **Write a model:**

As described above, write a model that takes in 8 inputs (The mouse's sensory matrix minus the center), and outputs four probabilities for each cardinal direction (N, S, E, W).

In addition, you will be handling when to back-propagate a reward, or when to keep running.

Additional task (Optional worth extra credit):

There are many variables that you can mess around with to complicate the problem to make it more suitable to Reinforcement learning. One of them is decreasing the range of the mouse's scent with the variable SCENT_RANGE. Another is the variable VARIABLE_TERRAIN which gives the mouse a second sense (sight) and adds a value to each terrain section which indicates how much energy is spend by stepping over that tile.

Brainstorming session 3:

Q: How does the reward function change if you add the new variables?