Introduction

**Decision Making Systems**

The foundation of the maze solving process is the combination of the two processes studied by Tolman and Hull; respectively the deliberative and habit systems. In early learning, a rat in a maze solving trial engages in the deliberative process, trying out various paths and attempting to determine at each crossroads which path will lead to the higher reward. (Redish, 199) Later in learning, however, the rats switches to a habit based action strategy, in which it develops response-chain habits and stops relying on the more computationally expensive deliberative system. (Redish 199)

The transition between these two decision making systems is clearly illustrated by Packard and McGaugh. Rats were tested in a simple T-maze, where a right-turn was reinforced. When they were tested after 8 days of training with a reinforce on the left side instead, rats successfully switched their plan and took a left turn. This action-outcome decision is reliant on the hippocampus, and not on the dorsal striatum. Furthermore, when the rats were tested after 16 days of training, the opposite results were seen: rats continued to make the right-turn even when the reinforcer was on the left, relying on the S-R association in the dorsal-striatum system and not the hippocampal system.

These concepts are further supported by Yin and Knowlton (2006). They showed that while win-stay training supported by the dorsal striatum is not affected by devaluation, win-shift performance that relies on the hippocampus is affected by devaluation. Thus, we can conclude that rewards resulting in dopamine bursts and dips are crucial in the beginning stages of learning but not for experienced foragers.

**The Role of Dopamine**

Dopamine cells signal novel events, resulting in either a dip in dopamine levels in response to a negative event or absence of expected reward and a rise in dopamine levels in response to a reward (Brown, Bullock, & Grossberg). Once the outcome of a response has been established, an expected value is determined for a particular response in the presence of a cue. In the case of an unchanging maze, the probability of a reinforcer being delivered or not at a particular location is 1. Therefore, once it is established whether or not the reinforcer is present, the rat can reliably determine the expected value of the location, in other words whether or not to return to it in the current or later trials.

Cells from the SNc and VTA send dopamine signals to the frontal cortex, amygdala, and basal ganglia, which are responsible for reward related learning. The dopamine signals are then able to strengthen connections between cue and plan representations and the basal ganglia will finally choose and execute the best supported plan. This competitive framework explains why in the beginning of learning actions are highly reward contigent, and can be altered by moving the reward to a different location. However, once the dorsal striatum habit system has been strengthened, it will be capable of overriding the hippocampal system and cause in some cases more efficient decision making, and in novel scenarios cause errors until old habits are weakened over time.

While the direct pathway increases dopamine levels, the indirect pathway disinhibits the SPi/SNr by decreasing acetylcholine levels which normally works to suppress action. The indirect pathway is also responsible for long term depression, through which an established habit can be extinguished through time when it is repeatedly not reinforced.

**Implementation**

The Maze Problem Solution program works by creating building a maze and allowing it to be explored, while accumulating information about the reward or lack of reward at visited locations. Its ultimate goal is to model the rat’s exploratory process and illustrate how receiving or not receiving rewards at different locations will lead to less exploration and a more direct route in progressive trials.

**Set up**

The maze is set up as a two-dimensional array with numbers set to represent different location values (0 = path, 1 = wall, 3 = dead end, 5 = start, 6 = end). This will create the graphic displayed for the rat to run through.

In order to represent the hippocampal place map which the rat establishes by associating locations with reward or no reward, another two-dimensional array is created. This array values all dead ends at -1 and the end of the maze at 1, while all unexplored territory has a 0 value initially.

The trial is set to run for 5 trials

For numTrials = 1:5

The current and previous locations of the mouse are stored in variables, with the mouse starting out at the beginning of the maze (2,2).

**Test Runs**

The actual test will begin within a “while loop” which will be broken out of when the rat reaches its final destination = 14,14.

Step 1: Display the maze

plotMaze(maze, expMap, curMouseLoc)

* Displays the image of the maze with walls, start and end point
* Iterates through the expectation map and where it finds -1 values, adds in a red dot in the corresponding image map location, doing the same for +1 with a green dot.
* Creates a circle image which will represent the mouse and places it at the 2,2 value in the image map.

Step 2: Compute all possible moves

* listOfMoves = possibleMoves(maze,curMouseLoc,prevMouseLoc);
* The X and Y locations of the mouse currently are passed into the function – for example from the starting position the two numbers passed in are 2 and 2.
* An array is created to hold the possible moves.
* The four possible locations (above, below, right, and left) of the current locations are tested to see if they satisfy the following parameters
  + Not the same as the previous location (either x or y is different)
  + Not a wall (equivalent to 1 on the map image)
* If a location satisfied those requirements, it is added to the array of possible moves. Note this number will always be 2 or less in the given maze since it cannot include the previous location and unless the mouse is at a crossroad will be equal to 1.

Step 3: Update the expectation map

* expMap = updateExpectationMap(expMap,listOfMoves,curMouseLoc);
* Before the mouse actually moves, it needs to be recorded whether the original decision to move to this spot was a good or bad one. If there is a positive reward adjacent, it should be updated to be positive, while if there are only negative rewards, it should also be labeled as negative.
* The possible moves array will be used to extract the possible rewards from the expectation map.
* If any of these possible rewards is equal to 1, the value for the current location will be increased to 1.
* If all of the possible rewards are negative, the value will be set to -1.
* Otherwise, no changes will be made since the artificially intelligent rat cannot draw a satisfactory conclusion on the goodness of its choice.

Step 4: Choose a move

* moveToLoc = chooseMove(prevMouseLoc, listOfMoves, expMap);
* Search through the expected values for possible moves
* If one of them has a value of +1, select that one.
* If all of them are negative, go backwards.
* Otherwise, randomly selected a move from neutral directions.

Step 5: Update locations

* Redefine previous location and current location variables to accurately represent the move.

These steps are repeated until the mouse finds (14,14).

The (-1) values in the expectation map are the key to the mouse’s ability to reach the goal faster in subsequent trials. By assigning -1 when a mouse reaches a dead end the program ensures that the mouse will not go to the same location again. It eliminates options and as it does so, leaves only a more direct path to follow. Additionally, with each trial it builds up knowledge about the steps it took to find the goal. +1 expected values build up during each trial, with one additional +1 being added on the end of the trail each time. In this fashion, the rat will eventually have a trail of +1’s leading from the goal to the start point. It will then continue to always follow the same route.

In earlier trials, therefore, the rat is relying largely on negative feedback, knowing which places he should avoid. It is possible that due to randomness, during these early trials a rat may approach the goal from different directions on different trials. However, as the +1 trail elongates, the rats process will become more solidified and one final path will be created with the rat will not deviate from in the future.

**3. Critique**

This maze model is an extreme simplification of part of the process that rats actually use to learn how to navigate a maze. The basic concept is an attempt to model the behavior learned in the ventral striatum and amygdala, by labeling each square in the maze as a “good” place or “bad” place based on past experience or the location of a dead end. The only response available to the independent agent (rat) is to approach or avoid a square based on its assigned numerical value (-1, 0, or 1). While this illustration is interesting, it could not be used to model actual rat behavior for various reasons.

First, it does not include input from the hippocampal system which would create a spatial map, and is what rats actually greatly rely on when first exploring a maze. Because of this rats have to try randomly selected directions after calculating the expected value of each option. They cannot try out various hypotheses (such as taking all left turns or right turns). This means that they would also not be able to compare the speed of various routes and may end the set of trials with a sub-optimal path.

The values assigned to each “good” square also do not accurately reflect dopamine levels or reward values. In reality, reward values would become devalued over time, and all the locations leading up to the actual reward would not express the same reward value. So, though the numbers function to lead the rat to the reward in the model, they are not actually representative of the motivational factors that act in reality.

Finally, rats could be made into better learners by leaving a trail of -1’S behind when leaving a dead end arm. In reality, rats often make decisions at branching off points rather than simply associating the last negative decision that they made with an adjacent square. By labeling squares only that are adjacent to negative ones, the rat can functionally by trial and error create a path that will lead to the final reward, however that path does not reflect how it would be represented in a rat brain.

The part I found most appealing about the model was its ability to accumulate learning over trials if iterated various times and use the final expectation map of the last trial as the base expectation map of the next trial. While the model is too simple to model real learning this component offers an interesting strategy for modeling learning over time, and with more components added to this model it could be useful for comparing different brain systems, and eventually combining the functionality of all of them to create an artificially intelligent rat that mirrored real learning more closely.