Comp 596 Assignment 2

Jonathan Pearce 260672004

April 22, 2020

Part 1

Problem 1a The distal reward problem addresses how to propagate goal information in order to improve behaviour across the entire environment. Most locations in an environment are far from the goal location and therefore most place cells are not (very) active when the goal is reached, meaning they learn little to no new information.

Problem 1b A navigation learning model that suffers from the distal reward problem is one where the model assumes that place cells are the optimal method for representation in a reward based learning environment. The distal reward problem is prevalent in this type of navigation learning model because only the place cell(s) that were active immediately before reaching the goal are able to associate with the action that brought the agent to the goal. Therefore place cells that are far from the goal will never receive a learning signal and the agent will have no information on what direction to move when placed far from the goal.

Problem 2a The global consistency problem addresses how to learn about the environment as a whole (globally) when only provided or only capable of obtaining self motion information relative to a trajectory origin and faced with unpredictable motions such as starting a trial in a random new location.

Problem 2b A navigation learning model that suffers from the global consistency problem is one where the model assumes that the place cells become associated with metric coordinates for locations within the environment and these coordinates are learned through the agent's self motion information. The global consistency problem is prevalent in this type of navigation learning model when the agent undergoes unpredictable motions, such as episode restarts in random locations. Integrating over self motion estimates starting at each new starting location leads to the agent learning inconsistent coordinates over the environment as a whole.

Problem 3a The place cells (modelled as Gaussian functions) are used to map the position of the agent in the continuous state space to a finite feature vector. The place cells encode the agent's location because their activation is dependent on where in the environment the agent is located

Problem 3b The place cells only encode the agent's location in the global coordinate system. A navigation system might want information about other spatial or navigational quantities such as distance and direction from a distant goal as well as information with respect to the agent's local coordinate system (i.e. with respect to where the agent began that episode).

Problem 3c The critic attempts to learn the optimal value function over the entire state space. Given the place cell activation for the agent's current location the critic should return the expected discounted total future reward the agent will receive (in the episode), assuming it follows the actions specified by the actor.

Problem 3d The actor attempts to learn the optimal action policy for the agent over the entire state space. Given the place cell activation for the agent's current location the actor returns a probability distribution (policy) over all available actions. The actor encodes how frequently each action should be taken at each position in the state space.

Problem 3e The TD error δ_t is calculated as follows,

$$\delta_t = R_t + \gamma C(p_{t+1}) - C(p_t)$$

Where R_t is the reward the agent receives at time t. $\gamma \in [0, 1]$ is the discount factor. $C(p_t)$ and $C(p_{t+1})$ are the critic's value estimates of the agent's position at time t and $t + 1$ respectively.

Problem 4a Yes, TD learning with an actor-critic system does solve the distal reward problem. When the goal location is found for the first time, the TD error and subsequently the actor and critic weights become non zero and they begin learning about the environment. The critic will begin valuing states closer to the goal higher than states farther away from the goal and therefore the TD error will be non zero and learning can take place at every time step instead of only when the goal location is reached and our agent receives a reward. Once $C(p) \approx V(p)$, the TD error will enable the actor to accurately reinforce actions that move the agent closer to the goal, and will suppress/discourage actions that move the agent away from the goal, regardless of how far in the environment the agent is from the goal.

Problem 4b In the paper the authors discuss that in order for a rat to solve the global consistency problem with TD learning it must be able to obtain instantaneous and accurate estimates of its self motion, in other words it must have dead-reckoning abilities and the rat must have input from place cells. More specifically for our task the authors motivate the need for a dead-reckoning system that is dependent on input from the place cells. This system learns an allocentric representation of the environment, which is independent of the animal's point of origin. This solves the issue of the rat being placed randomly in the environment at the start of a new trial. I would argue that a rat would possess these two pieces of information. In the paper the authors cite another work which demonstrated rats dead-reckoning abilities. Rats also have hippocampal place cells which fire when a rat occupies a restricted portion of an environment

Part 2

NOTE: In the multi-platform experiments I have defined the locations, as opposed to generating them randomly. The first four locations are each in one of the four 2D quadrants to ensure the agent is tested across the entire environment.

Problem 1 See submission notebook

Problem 2 With a single platform location, the actor-critic model successfully reduces the latency over trials (Figure 1). The performance of my actor-critic solution is not quite as efficient as the model in the paper. A more extensive hyperparameter search could potentially lead to equivilant results.

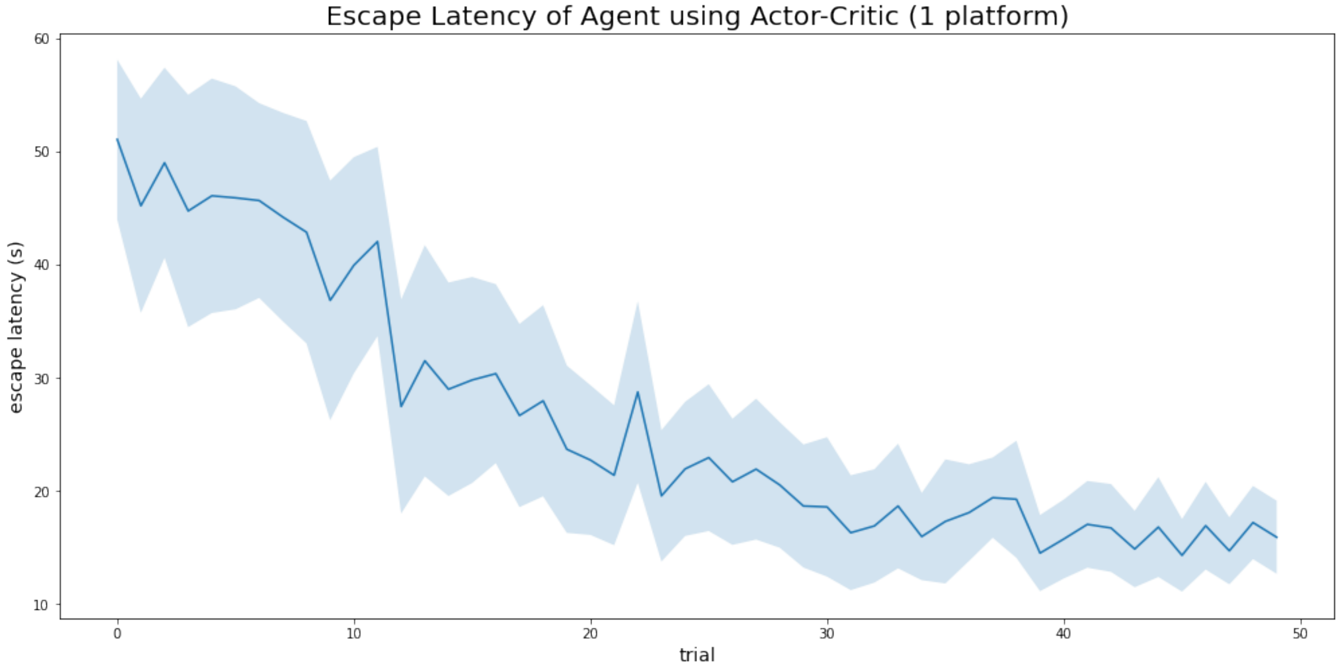

Figure 1: Performance of Actor-Critic using TD learning on Watermaze Environment with one platform location. Averaged over 50 independent runs. Actor learning rate = 0.1, critic learning rate = 0.01, discount factor = 0.9. Platform location: (25,25).

Problem 3 When the flag in my code is toggled to the multi-platform case the agent struggles to reduce the escape latency for the second platform location (when compared with the first platform location). This inability to quickly reduce the escape latency continues for the third, fourth and fifth platform locations (Figure 2).

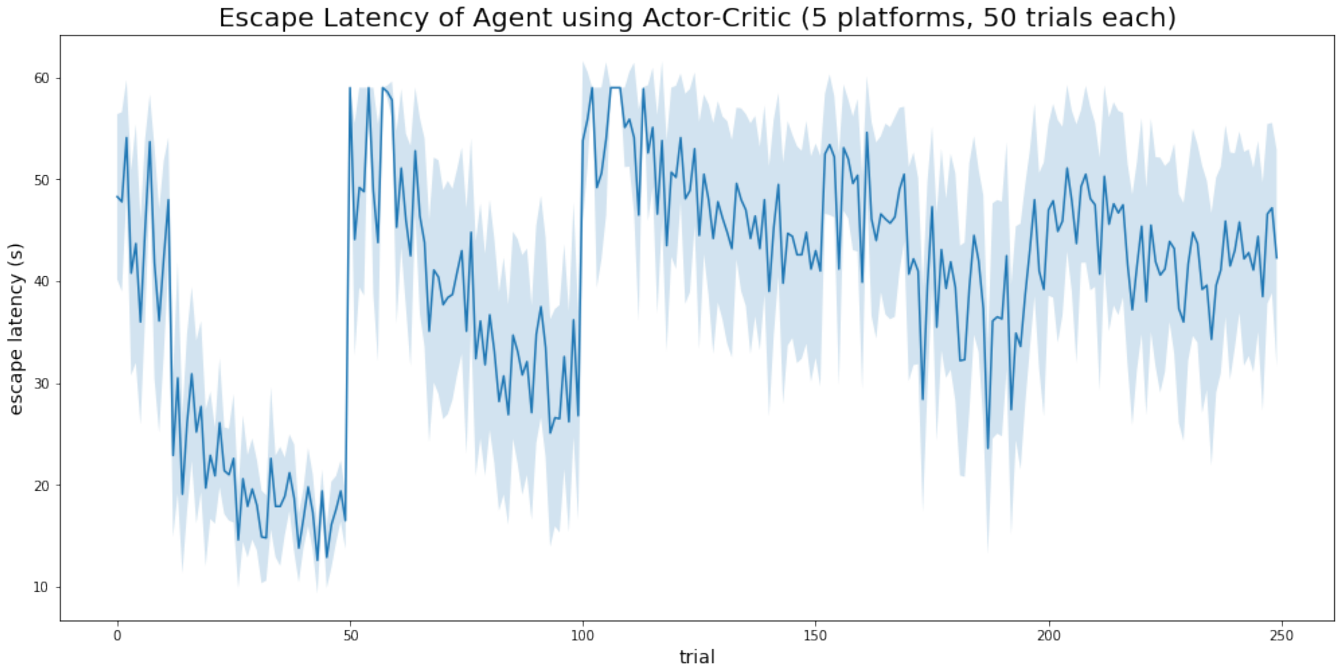

Figure 2: Performance of Actor-Critic using TD learning on Watermaze Environment with five platform locations. Averaged over 10 independent runs. Actor learning rate = 0.1, critic learning rate = 0.01, discount factor = 0.9. Platform locations (in order): (25,25), (-10,-15), (-20,25), (15,-35), (40,35).

As discussed in Part 1 Question 4a the actor-critic framework with TD learning is able to solve the distal reward problem. Therefore for a fixed goal location the actor-critic method from the paper is successful at quickly reducing escape latency over time (Figure 1). However, when the goal location is changed the actor-critic method fails to reduce the agent's escape latency as quickly for two key reasons. First, the actor-critic method fails to recognize and compensate for the fact that the learned value function and policy from previous platform locations interferes and slows down the actor-critic system from adapting to a new platform location. Looking ahead, this interference problem is solved by the coordinate action in the combined model, which when chosen by an agent with a goal memory, moves the agent directly towards that goal memory regardless of learning in previous trials. Second, the information learned by the value function (critic) and policy (actor) cannot transfer when a new goal location is introduced. When the actor critic system has learned about a particular platform location well and the platform location changes, The actor critic system must completely re-learn the value function and policy. Again looking ahead, this issue is addressed by the coordinate learning model which seeks to learn globally consistent position approximation using place cell activations. This method is not significantly effected by changes in goal location since it is attempting to learn global coordinates of the environment.

Part 3

Problem 1 See submission notebook

Problem 2 With multiple platform locations, the combined model successfully reduces the latency over trials (Figure 3). The performance of my combined model solution is not quite as efficient as the model in the paper, as we do not achieve one-trial learning. A more extensive hyperparameter search could potentially lead to equivilant results. Figure 4 in my report is a replication of Figure 6e from the paper and shows that my combined model quickly becomes totally dependent on the coordinate action, similarly to the results in the paper.

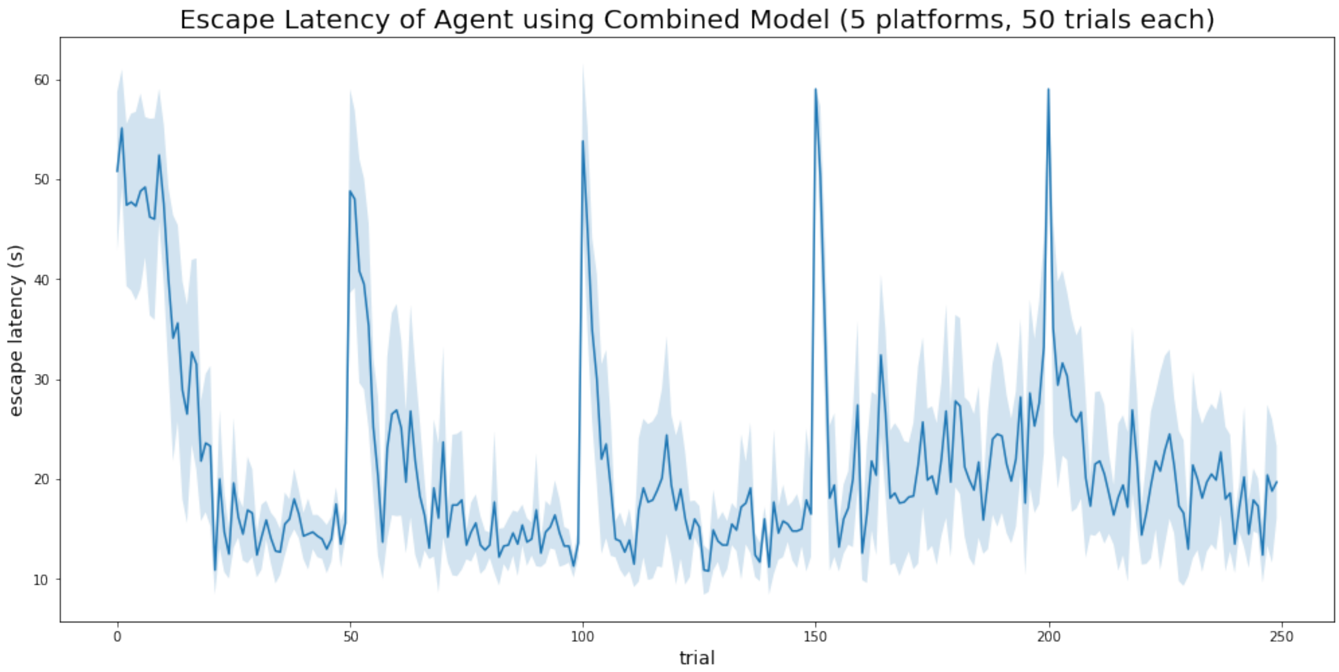

Figure 3: Performance of combined coordinate and actor-critic model on Watermaze Environment with five platforms. Averaged over 10 independent runs. Actor learning rate = 0.1, critic learning rate = 0.01, discount factor = 0.9, coordinate model learning rate = 0.001. Platform locations (in order): (25,25), (-10,-15), (-20,25), (15,-35), (40,35).

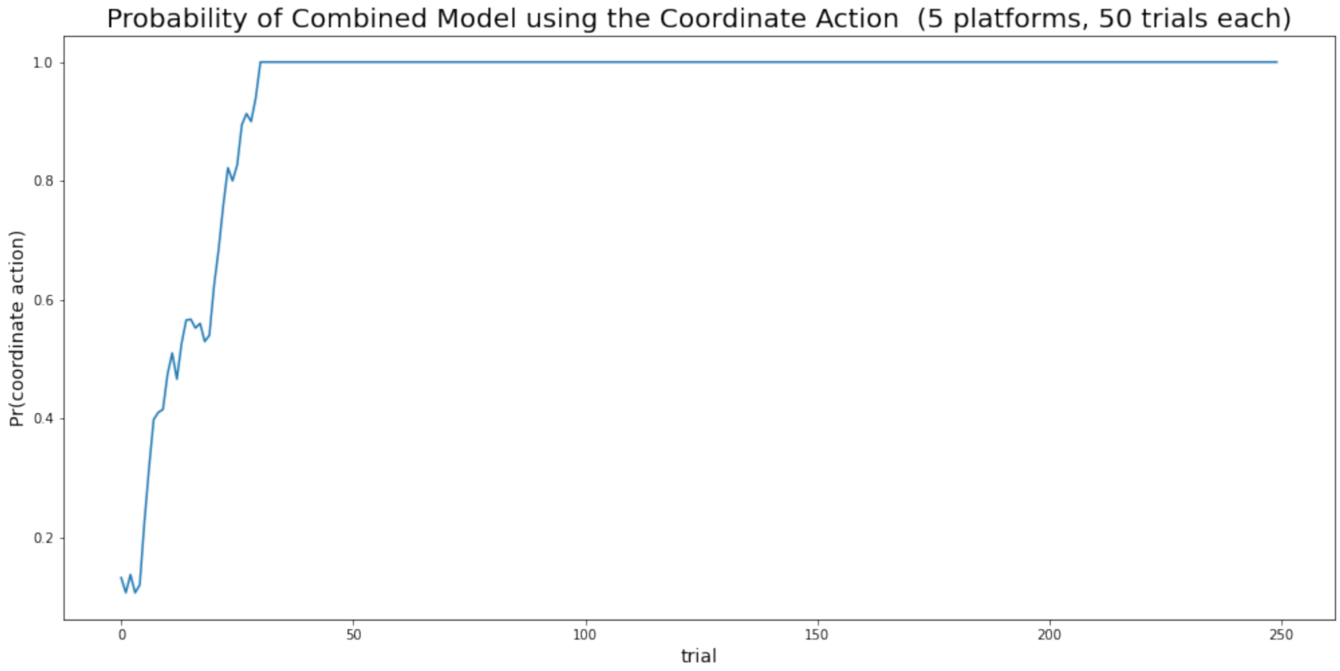

Figure 4: Likelihood of combined model selecting the coordinate action. Averaged over 10 independent runs. Actor learning rate = 0.1, critic learning rate = 0.01, discount factor = 0.9, coordinate model learning rate = 0.001. Platform locations (in order): (25,25), (-10,-15), (-20,25), (15,-35), (40,35). This is a replication of figure 6e from the paper.

Problem 3 In this experiment our agent's environment is a 2D space with no barriers. This simplistic environment is what makes the coordinate model specified in the paper work so well. In the watermaze task there is always a direct path from the agent's position to the goal location and therefore the greedy action direction that the coordinate model specifies (when a goal memory is present) works incredibly well as demonstrated by the few trial learning in my experiments (Figure 3). However, the coordinate model's capabilities would be immediately reduced if barriers were added to the environment. If a barrier came between the agent and the goal location, the coordinate model's greedy actions that go directly towards the goal memory would not be optimal. A more technical way to consider this idea, is that the coordinate model seems to only work effectively in convex spaces. This idea can be further expanded if we consider a higher dimensional space, for example a more complex surface, with elevation changes. If an agent were trying to navigate to a goal location while changing elevation as little as possible (to save energy), an actor critic model with TD learning would be able to learn how to avoid hills and valleys to minimize energy usage, the coordinate model would simply move directly towards the goal without considering the change in elevation. Therefore in simple convex settings, the coordinate model appears to be a good choice for an AI system. However, for more complex spaces, surfaces and tasks the coordinate model is far from optimal and should not be choosen for an AI system. Further, the coordinate model seemed to be very sensitive to hyperparameters further discouraging its use for an AI system.