CS-370 7-3 Project Two

Cade Bray

Cade.Bray@snhu.edu

Southern New Hampshire University

Olga Mill M.S.C.S, M.E.M

February 23rd, 2025

During this project I have implemented a deep Q-learning algorithm into a Non-Player Character (NPC). Its objective is to navigate a map that is an eight-by-eight grid with impassable objects. The intent for this pirate is to obtain the treasure located on this map before the player does. Analyzing the differences between how a human and a machine approaches this problem can be very insightful. Starting with how a human would solve this maze would be to navigate each square from left to right or top to bottom until it finds the objective effectively using trial and error. A machine uses a similar method during its training and will at first be purely trial and error finding its treasure. Once an area is known to be a dead end a person will avoid the area because they will quickly remember the consequences of its actions. The machine may continue to retry until it's adjusted its weights to know there isn't a reward for following this given path (Beysolow II, 2019, p. 4). The more layers a model has the quicker it can make this association as well.

The difference between exploitation and exploration may not be apparent in a human's actions immediately but in a machine, it plays a critical role. Every time a machine is faced with a new decision it will calculate whether it should explore new options or pick a previously known rewarding option to follow. As the machine becomes more successful it will become less inclined to explore because it is becoming more confident in its ability to succeed through this path (Beysolow II, 2019, p. 71). This is achieved by using an epsilon factor which is calculated between zero and one. Every time the machine goes to determine exploration or exploitation it will generate a random decimal point between zero and one and compare that to what epsilon is set to currently (Gulli & Pal, 2017, p. 278).

Knowing how exploitation and exploration works we can explore the optimal balance between these two choices. If a machine only chooses exploitation from the beginning or is more inclined to exploit known paths, it will take considerable time to discover new and rewarding avenues. Whereas if a machine defaults to over-exploring it will become flooded with improper movements to choose from in its memory (Gulli & Pal, 2017, p. 270). The optimal decision making is to explore early in a machine learning process and slowly reduce that exploration as it becomes more successful in its actions thus retaining more exploitative movements in its memory (Beysolow II, 2019, p. 85).

Reinforcement learning can help guide a machine to more successful paths by using rewards and penalties. If a machine successfully follows a path to treasure, it will update each previous step it took as a positive weight to incentivize it to take that path again. Following these incentives is what happens during exploitation. The machine will consider all available actions and determine which has the highest reward . When the machine discovers a dead-end and losses the game it will penalize each step it takes to ensure that the machine is less likely to repeat those actions. Eventually paths that always lead to treasure will have an overwhelmingly large reward associated with it combining with slowly taking the exploration route it will continue to take known rewards to success (Gulli & Pal, 2017, p. 268).

In my Q-Learning implementation the epsilon will be lowered by 0.5% every epoch that is above a 90%-win rate. Eventually lowering epsilon to a chance of exploring at 5%. The reason I do not let epsilon decay completely to zero is because it will close off any potential doors to improvement over the course of its training. This means that if a shortcut that would later be revealed is not discovered before total epsilon decay, then it never will because the machine will always choose exploitation and follow previously rewarding paths. Additionally, I have balanced the epoch and batch sizes to create an optimal environment for training (Bray, 2025). Distinct factors can impact how long a training session goes on for in a model. By decreasing the epoch, you are reducing the overall training time. In the epoch we can observe a model fitting call that will run four epochs once all the data is gathered. Each epoch helps converge on an optimal weight distribution revealing the solution to the treasure hunt game for that specific state and action.

References

Beysolow II, T. (2019). *Applied Reinforcement.* Apress.

Bray, C. B. (2025, February 23). CS-370 Project Two Jupyter Notebook [Bray_Cade_ProjectTwo.Ipynb].

Gulli, A., & Pal, S. (2017). *Deep Learning with Keras.* Birmingham: Packt.