CS370 Design Defense

There are some glaring similarities between an intelligent machine learning system and the way humans learn. After all, let’s not forget the machine learning design is modeled after the reinforcement learning of mammalian brains. Both the human brain and AI systems function off of a reward-based architecture. For our pirate maze as an example, if a human were operating the pirate’s actions, sort of like a computer game they would also enter into a sort of guess and check method. However, the major advantage is the birds eye view that we typically are given when completing a 2D maze. The pirate agent in our AI system is doing the task blind, so to speak. If I were perhaps in a 3D maze, my action and that of the pirate agent would then be very similar. Just like the machine learning software, the human would trial new directions to see if they are available or not. Just like the system, the human would be rewarded if their attempt turned up an available space to continue. One difference I can imagine is how the AI system would have a much more accurate memory database than does your typical human brain. So the human might forget what routes they have already tried, resulting in repeated attempts at the same wrong path. The AI system would recall as much as you set their memory to.

There are two main searching methods carried out by the intelligent system in order to complete the maze. These are known as exploration, and exploitation. Exploration, as the term suggests, involves the agent sort of blindly trying a novel action, or in our example trialing a square it had not yet. The following reward, or lack thereof, is then logged in the systems memory. This historical database of past trials and their rewards is what is involved with exploitation. Exploitation is when the system relies more on its memory to make an informed decision of what square to trial next. Both exploration and exploitation are used in this system, and it is always important to trial different proportions in order to determine the best fit for solving a problem. For my system, the exploration is a higher proportion than exploitation. This is because we need the system to explore more than to recall, as it is searching for an undiscovered route. However, it is important not to have exploitation set too slight, as this can lead to increased time to solve due to repeated attempts at occupied squares. Reinforcement learning applies these two searching methods, and couples them with a rewards system. All of this data is then stored in the memory for recalling what actions received what rewards, and just like us the system begins to learn through positive reinforcement, or reinforcement learning.

At a more detailed level, we can observe the inner workings of these so called neural networks, and see how these algorithms provide the framework for reinforcement learning and solving complex problems. These can be thought of as mathematical equations. Each neural network is comprised of an input layer, hidden layers, and an output layer. The input layer is made up of one or more variables, or information, to be passed through the neural network. Next, the hidden layer or layers involve one or more neurons that process the information and determine true or false, and decide what neuron to transmit to in the next layer. The more layers and more neurons in the hidden layer, the more processing and more accuracy. Lastly, the output layer consists of the output variable produced by the processing (Shin, E., 2020).

These algorithms involved in AI systems are already widely utilized to solve complex problems, and their application domain is growing. From personalization to guess what someone might like to watch for entertainment, or read on their news feed, to smart cars and flight systems, our modern world is becoming increasingly embedded with the use of AI.

I implemented DeepQ learning with neural networks in an attempt to solve the pirate maze problem. By capturing the game experience data, and utilizing exploration and exploitation techniques, the pirate agent is supposed to complete attempts, or epochs, of trialing maze navigation. I created a “for” loop to represent each epoch, which entailed the agent making a random choice at a cell. Then there is a “while” loop nested within this “for” loop in order to continue the searching, because the while entails the random cell chosen previously was an empty cell, and it is NOT game over, yet. The data from each episode, such as the action taken, reward gained, and game status, is all stored by calling the “remember” method and adding this episode to the systems memory. If the pirate turns up an empty cell, the while loop and data collection continues.

References

Shin, T. (2020, June 2) “A Beginner-Friendly Explanation of How Neural Networks Work: Understanding Neural Network Fundamentals for Five-Year-Olds”, *Towards Data Science*. Retrieved from <https://towardsdatascience.com/a-beginner-friendly-explanation-of-how-neural-networks-work-55064db60df4>