Programming Assignment 3 - Tic Tac Toe

By Nicole Kurtz

For Programming Assignment #3, I implemented a Q-Learning Tic-Tac-Toe player who use chose actions using on-policy epsilon-greedy action selection. To implement my Q-Learning, I created a HashMap where the string of the board is the key which maps to a Q-Value and list of actions for this board state. I initialized each Q-Value to 0 when it was added to my hash map. It is important to note that I always had my Tic-Tac-Toe agent play the game first, which gives an advantage.

I trained my agent on 5,000 games over 200 epochs. After the completion of each epoch, I ran another 10 games to test the current performance of the agent. At the beginning of each training epoch, I reset the value that represented the probability of exploring to 0.8. A larger number means that the agent is more likely to choose a random action over an action with the highest Q-value. During a training epoch, I reduced this probability after a consistent number of episodes. Additionally, I reduced the initial probability value after a set number of epochs to reduce exploration after enough training.

I chose 0.9 for my Learning Rate with a 0.75 discount factor. When my player lost, I gave a reward of -1. When my player won, I gave a reward of 1. And lastly, when my player tied, I gave a reward of 0.5.

My results are shown in the graph below titled “Test Win Percent after Training Epoch”. As you can see, after about ~150 epochs my agent performed consistently between 90-100%! Even after about ~5 epochs, my agent performed typically between 90-100% with the occasional dip to as low as 80%. Overall my agents win percent over 200 epochs was between 96-97%!