

Alex Hall

UMBC ID: Hall9

CMCS 341

Project 4 - Tic-Tac-Toe

Extra Credit

2. Saving configs.txt

The file format I used is quite simple. With each board is there is 5 values that need to be stored. The first and the most important the lowest serialized representation of the board (this value representation 8 boards, all of which are symmetric), the next is the number of times that board was played. The next value is the wins made with that board, and then the next value is the losses with that board. Lastly is the weight (which is really only $\text{weight} = \text{losses} - \text{wins}$). Here is an example of an entry into the “configs.txt”

Example of entry: 6 32023 27990 2537 -25453

You can see “6” is the binary representation of the board, “32023” is the time played, “27990” is the number of wins, “2537” is the number of losses, “-25453” is the weight.

4. Symmetry

The way I did symmetry was by using a recursive function that would rotate the 3x3 array and add it to a list of symmetric boards. I also would mirror the board and recall the recursive function to get all the mirrored symmetric boards. Then with all 8 of the symmetric boards I find which of those has the lowest serialized representation and use that to now representation all 8 of those boards. I use that lowest serialized

representation to as the hashing value. In other words, my hashing function can now map $8 - 1$. What this means is if I have any of those 8 boards config they all map to the same hashing value and data, that means wins, losses, ties, probability and weight are all the same. This not only helps to reduce the entries in the hash table just it learns faster too. How fast might you ask? Well, it learns about 8 times as fast. Needing only to see a one of the 8 moves to generate the 7 other moves. By these methods it takes less moves to really learn how to win.

With this Symmetric $8 - 1$ mapping to the hash table it greatly reduces the number of entries within the hash table. In fact, the number of entries in my hash table is reduced by a factor of 8, as apposed to not using symmetry and not $8 - 1$ mapping which learns only by each individual move, while also hashing each individual move too.