http://www.informatik.uni-trier.de/~fernau/DSL0607/Masterthesis-Viergewinnt.pdf

http://fabpedigree.com/james/connect4.htm

<https://roadtolarissa.com/connect-4-ai-how-it-works/>

https://towardsdatascience.com/creating-the-perfect-connect-four-ai-bot-c165115557b0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Two unblocked 3-in-a-row's that don't require the same 4th cell cannot be countered, you will win next turn (unless your opponent wins first). This should also be incorporated into your heuristic, but may already largely be taken care of (except for a different 4th cell requirement) by simply counting the number of 3-in-a-row's (subject to the below). In general, 2/3-in-a-row's requiring the same cells to get to 4 as another 2/3-in-a-row isn't all that helpful.

You also need to count disjoint sets, e.g.:  
xx x can be thought of as a 3-in-a-row.  
x x can be thought of as a 2-in-a-row.

You also need to exclude 2/3-in-a-row's that can't get to 4, e.g. oxxxo contains a 3-in-a-row, but it's surrounded by opponent-occupied cells, thus you shouldn't count it.

You also shouldn't count a 3-in-a-row as a 3-in-a-row and 2 2-in-a-row's. Since you didn't give your functions, I can't tell if you're doing this.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Currently, to the compare non terminal states, the computer counts the number of open 3 in a rows each player has. This simple rule allows the computer to understand the previously inscrutable board state like this instead: