Milestone 2

Fuzz in a row

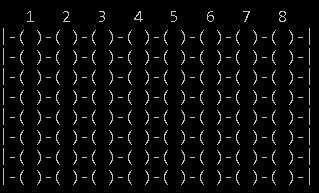
Stijn Verdenius 10470654

**Goal/research question:**

The goal is to create an AI agent for connect-4 that is significantly better than an AI based on the a Brute-Force(3 deep) algorithm and an AI based on a neural network. Since these are deterministic we only need to play 16 games each (2 for each player starting first and 8 for each first move) to determine who outperforms who. I endeavor to research the application of fuzzy logic in discrete decision making.

**Setup:**

The game consists of two players, each gets to throw a single disk into one of the 8 columns in their turn, after which the disk drops down as far as it can. The player with 4 in a row wins.



The setup is operational and so are the AI- opponents, therefore the fuzzy AI could play the others once it’s ready.

**The Board:**

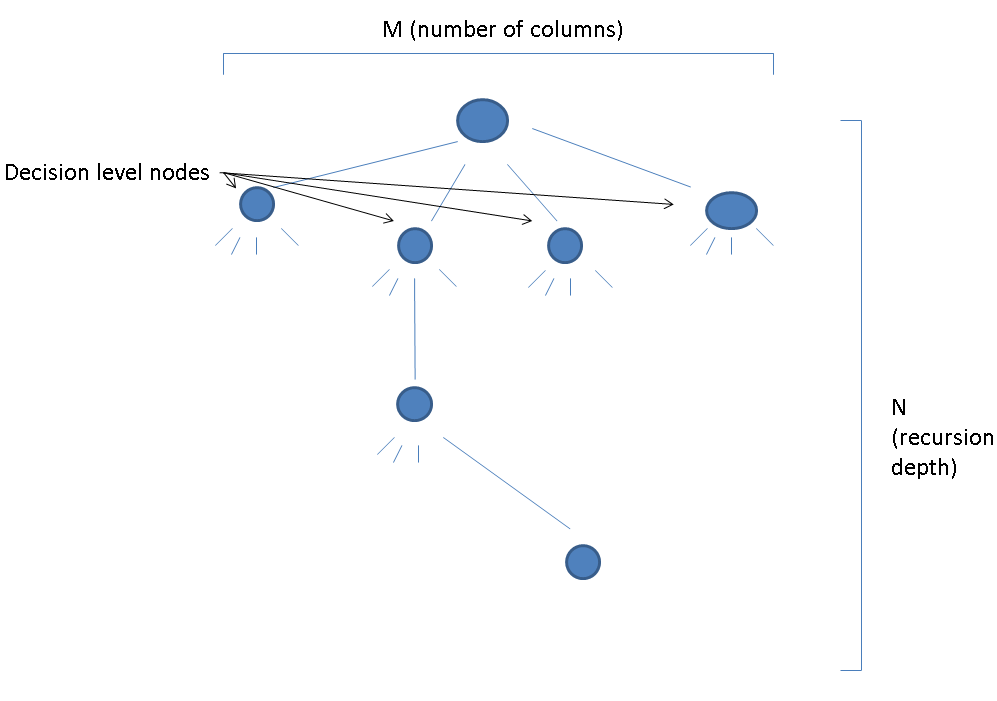
The traditional connect-4 board (6x7) is solved (Allis, 1988) and therefore I use an 8x8 board.

**Plan:**

As you have to get a discrete crisp output to make a decision in this game, it would be nonsense to use the columns as output: it will most likely occur that one would get float values such as 7.5 (where do you throw your disk then?) and the range is not continuous (column 3 might be better than column 5 but worse than column 4).

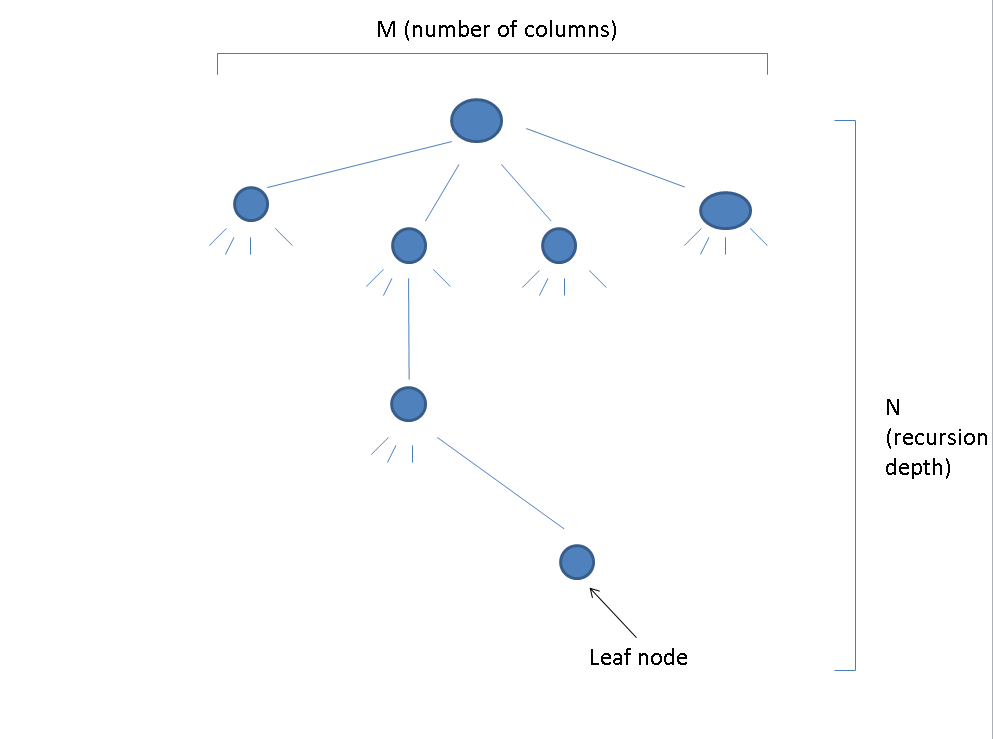
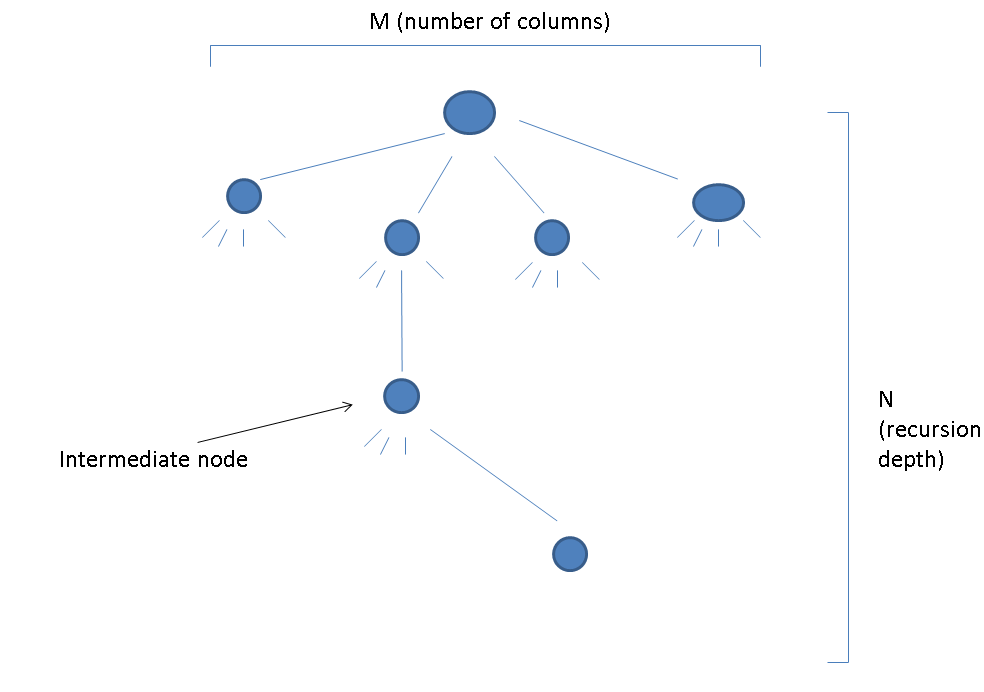
Instead, I will evaluate each of the possible moves on how brilliant of a move it is. So each column will get a grade (1-10) on how that puts the AI in more or less of an advantage and that will be the crisp output.

The crisp inputs will be variables defining the situation after a disk is thrown into a column (fuzzy controller 1). The crisp output for own-gain and opponent-gain will then be combined in a second fuzzy logic system that will give the final grade of a move (fuzzy controller 2). Then finally, the AI will throw the disk into the move with the highest grade (decision-nodes).



Of course the evaluation (fuzzy controller 1) of the move does not have to happen at one move down. We can recursively move down a decision-tree and let fuzzy controller 1 do the evaluation in the leaf nodes, passing the results up through the parent intermediate nodes until the decision nodes. Herby we can create a strategic forward thinking.

.



**Variables:**

For variables I count the number of occurring of certain patterns on the board. These are patterns I believe lead to a state of victory.

1: The first one is the number of potentials. These are single moves with enough space around them to eventually make a connect-4. An example:



2: The second is the number of blockings. This is any combination of three of the opponents disks, and one of the player itself. In this example, X is blocking O:



3: The third is the number of Win-In-Twos: that’s the amount of possibilities that could lead to a victorious situation within 2 moves. This is an example of a Win-In-Two for O.

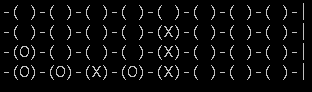


4: Logically follows is the Win-In-One, which is a win in two but then for one turn: example for X:



Variable 5,6,7,8 are the deltas of the last four, meaning: the amount the absolute values of the variables above have increased or decreased since the last move.

And finally we have the amount of moves made in the game so far (game progression), because we want the fuzzy Agent to pursue a victory as quick as it can. In this example the game progression is 8.



**What’s the fuzz?**

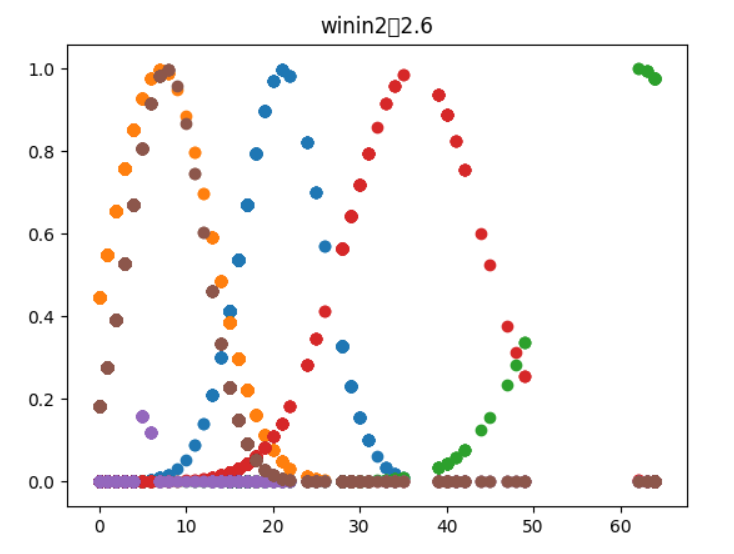
A question which might be raised by this point is: why fuzzy logic when this is clearly a discrete decision problem. The fuzziness is of a more linguistic interpretation of the state of the game. How good is the state the game is for me? And how good is it for the opponent? Is 15 Win-In-Twos ‘slightly good’, ‘pretty good’ or ‘amazing’. With fuzzy logic we can give expression to these uncertainties about the state of the game.

**Generating dataset:**

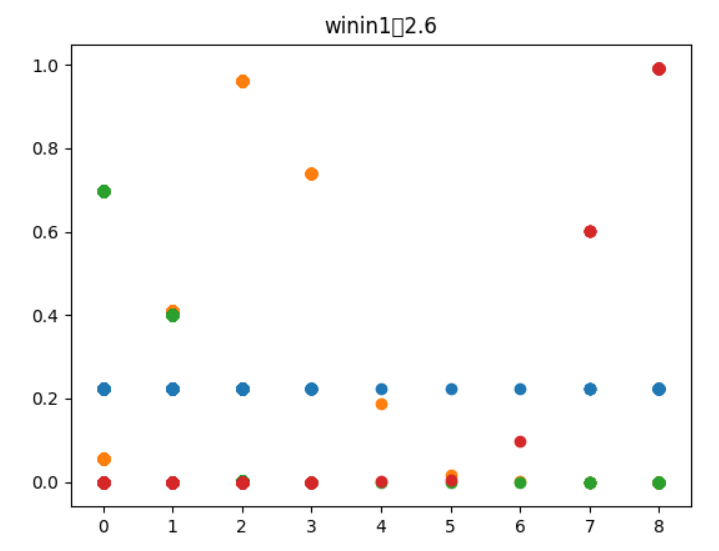
In order to learn from example the system is tweaked on a dataset. The dataset is taken from the evaluation another connect-4 AI player (a 3 layer deep brute-force one) and its evaluations on states give the output for certain variables. I have let the AI judge 200 games with its evaluation resulting in roughly 2,500 datapoints (one for each move). Consequently the dataset can be used for learning in the fuzzy controller

**Learning memberships:**

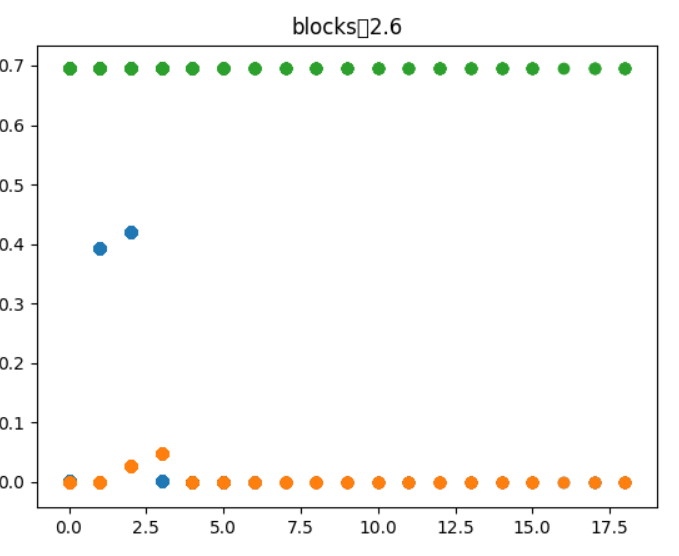
For learning memberships, we will use the c\_means method and then try to fit membership functions to the resulting membership degrees of the clustering algorithm.



Some variables however may have more trouble, due to the more discrete characteristics and consequently all the datapoints are roughly in the same place. An example of such is win-In-One:



Or there is just no clear function to be fitted. An example of such is the number of blocks:



In these cases some manual adjustments on the membership functions may be required, or it may even be necessary for complete variables to be taken out.

**Learning rules:**

When the memberships are determined we can learn rules with the wang-mendel (Wang & Medel, 1992) method. These rules are then saved to a fis-file which will enable the learned functions and rules to be opened in Matlab to inspect the surface-function. Once some manual tweaking’s have been done, the same fis-file can be used to transfer it back to python structure and the fuzzy-agent is ready to take it up against some opponent.

**Hypothesis:**

I suspect it will do better than the formal agents because the current results are already promising. Also, the recursion will give the AI an advantage over most humans, or the burte-force-3 of which it learns.

**Difficulties in the team:**

The “we” has turned in to “I” in this project, as one (Harm) of our teammates dropped out early because he found himself disinterested in the course and the other (Kaj) has overcome an intense illness. The problem is that while waiting for Kaj to recover, the work on the project has stopped and nothing happened for 2,5 weeks whilst I was waiting for him to get better. This ended when I got the news on 9th December that he was going to drop out, which is understandable but leaves me to do the project alone with 2,5 week delay.

**Journal:**

https://github.com/StijnVerdenius/Connect-4/blob/master/reading%20material/own/Journal.md

**Github:**

<https://github.com/StijnVerdenius/Connect-4>

**References**

Allis, L. V. (1988). *A knowledge-based approach of connect-four*. Vrije Universiteit, Subfaculteit Wiskunde en Informatica.

Wang, L. X., & Mendel, J. M. (1992). Generating fuzzy rules by learning from examples. *IEEE Transactions on systems, man, and cybernetics*, *22*(6), 1414-1427.