Playing Draughts using Neural Networks and Genetic Algorithms

Thien Nguyen

Department of Computer Science
Durham University

January 16, 2018

Outline

Problem Description

Motivation

Related Work

Approach

Curent Progress

Conclusion

Problem Description

A problem in Computer Science

Presently, competitive Draughts AI players are currently designed to play at a fixed ability.

While it has produced very competitive and intelligent players, they require manual modifications in order to improve its performance.

This is due to their dependency on pre-defined move databases, where optimal moves are pre-calculated, and recalled when necessary.

By combining Neural Networks and Genetic Algorithms, this issue could possibly be solved by creating a player that can grow in ability over time, without the dependency on move-banks.

Motivation

Why have I chosen to tackle this?

- Enjoyed the AI Search module
- Want to learn about Machine Learning (unfortunately not an option this year)
- I love board games!

Related Work

Similar works of art but no cigar

Samuel (59')

Uses Genetic Algorithms to improve coefficents of a set of heuristics to evaluate Draughts games.

Blondie24 (97')

Uses an Evolutionary Algorithm and Neural Networks to evaluate Draughts games. (Quite similar!)

Giraffe (15')

Uses contemporary machine learning techniques to train a Neural Network to evaluate Chess games.

Current Approach

How will I tackle this?

- Evaluate Checkerboard
- Choosing the Best Move
- Generating Agents
- Choosing the best Agents
- Making better Agents
- . . .
- Profit!

Checkerboard

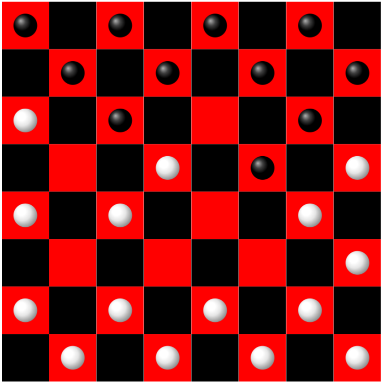
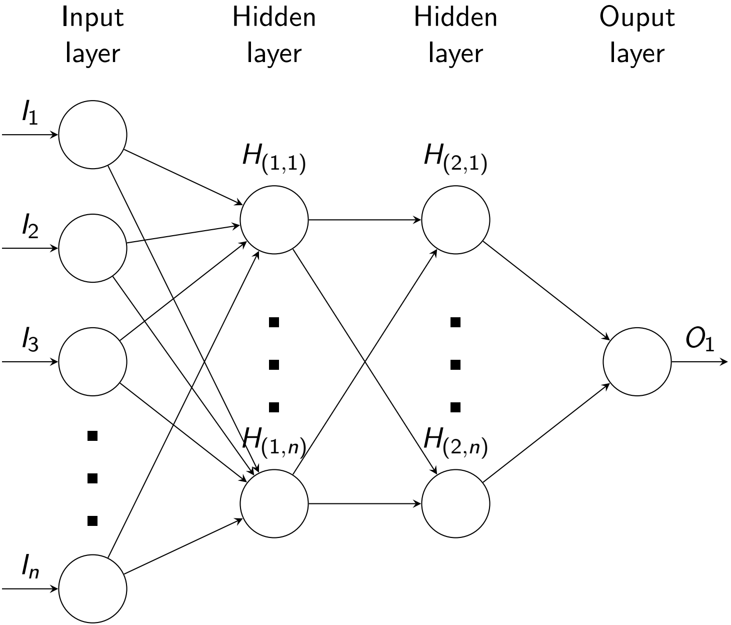


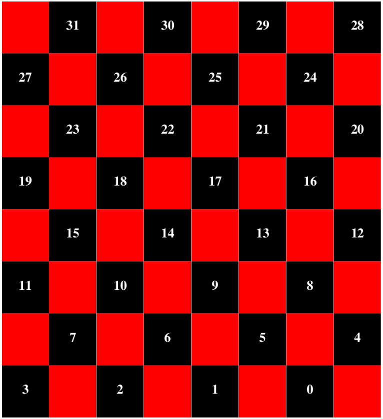
Figure: The indexes of the 32 pieces of the input layer are the immediate values of the positions on the board.

Neural Networks



$$Output = (Input + weight) * ActivationFunction + Bias$$

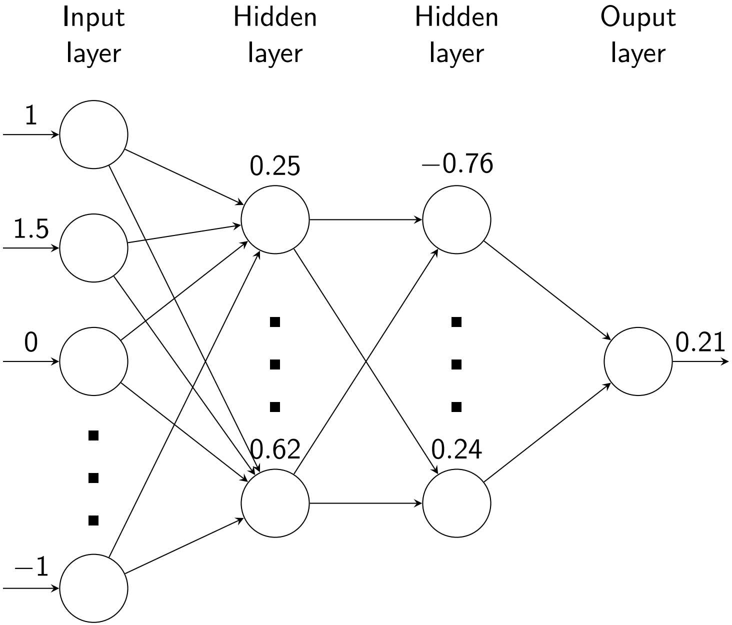
Checkerboard



	31		30		29		28
27		26		25		24	
	23		22		21		20
19		18		17		16	
	15		14		13		12
11		10		9		8	
	7		6		5		4
3		2		1		0	

Figure: The indexes of the 32 pieces of the input layer are the immediate values of the positions on the board.

Neural Networks



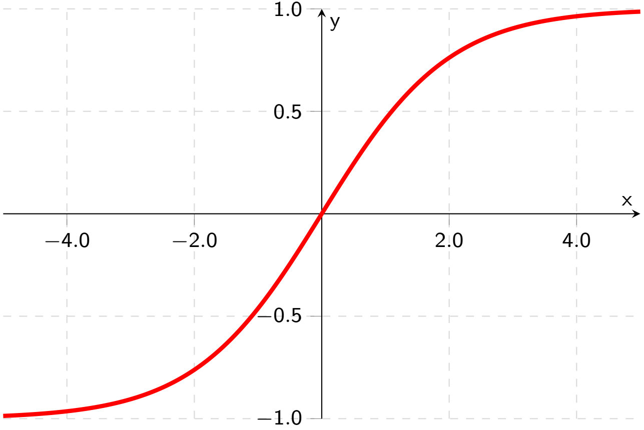
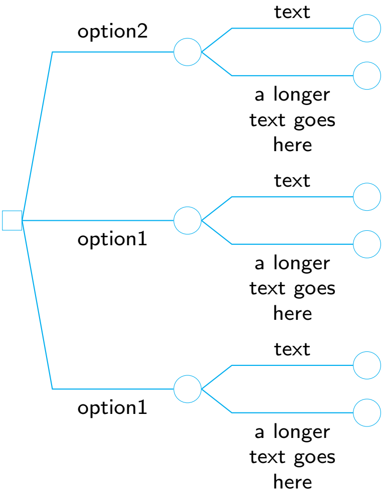


Figure: graph of modified sigmoid function $f(x) = \frac{2}{1+e^{-x}} - 1$

Choosing moves



Generating Agents

An agent is a generated set of weights for the neural network.

Tournament

1. Generate a population of random agents
2. Make agents play each other
3. Order agents by the amount of points scored
4. The best few agents are chosen to stay on for the next tournament
5. make new agents from those best few
6. the losers are destroyed
7. repeat step 2-6 with the new agents until satisfied

Crossover Mechanism

Mutation

Template

Current Progress

What have I done already?

- I've created a relatively ok AI bot.
- It plays relatively well!

Remaining Work

What do I still need to do?

Conclusion

What will I accomplish?

I will hopefully accomplish something.

References

Nanos gigantum humeris insidentes

References!