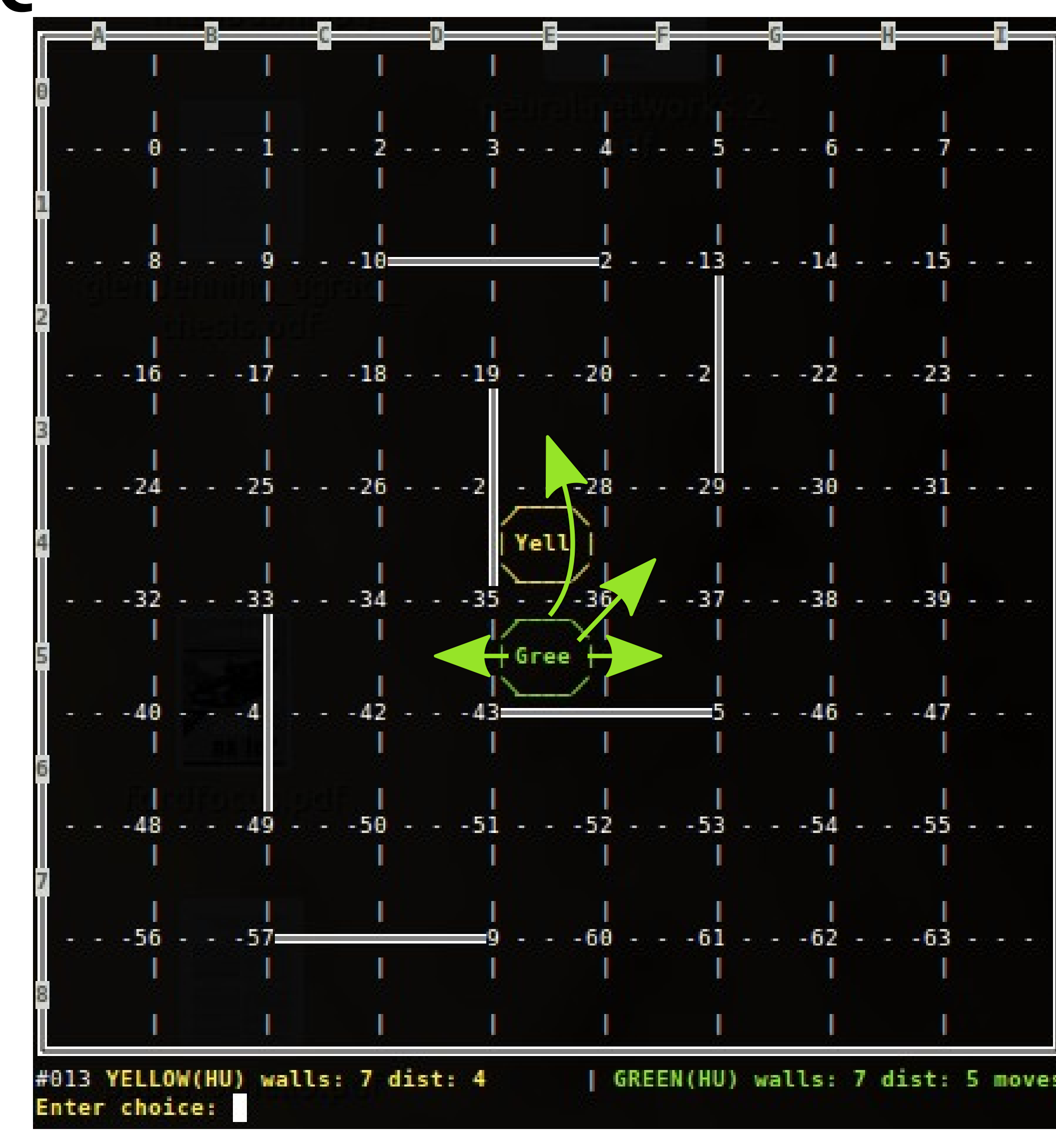
Neural Network as an Opponent in Quoridor

Advisor: Peter Gergeľ

1. Brief overview of existing approaches to the production of agents playing board games
2. Program intelligent agent built based on neural networks, which will learn to play Quoridor
3. Test end evaluate the behaviour of the agent

Neural Network as an Opponent in Quoridor

- 2 player version
- 10 walls each
- each starts in the middle of opposite sides
- goal is opposite side



Neural Network as an Opponent in Quoridor

State complexity

$$S = 2 \cdot S_p \cdot S_f = \mathbf{2.2775 \cdot 10^{27}} \neq 3.9905 \cdot 10^{42}$$

Branching factor

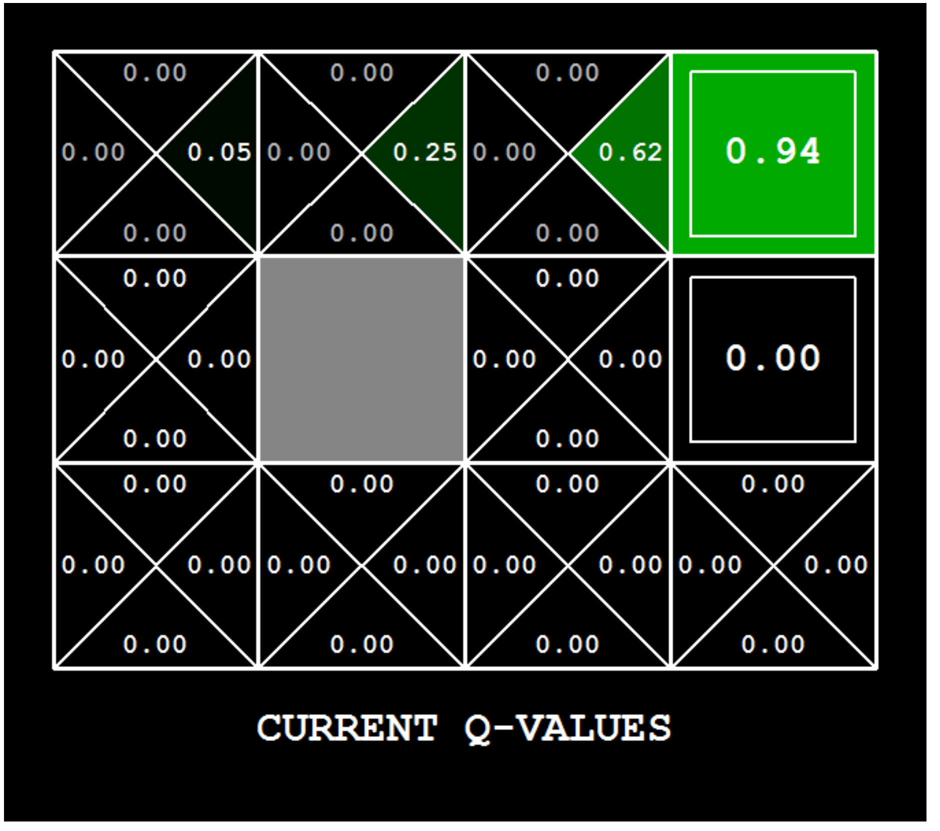
$$b = \frac{D}{S_d} = \frac{1.0766 \cdot 10^{28}}{5.6379 \cdot 10^{26}} = \mathbf{19.0974} \neq 60.4$$

Game tree complexity

$$C = 19.0974^{91.1} = \mathbf{4.9758 \cdot 10^{116}} \neq 1.7884 \cdot 10^{162}$$

Neural Network as an Opponent in Quoridor

Neural Network (NN) learns to estimate Q-value. NN's ability to generalize should help propagate Q-values faster to the early stage of the game.



Alpha is 1, NN handles learning rate

$$\hat{Q}^{new}(s_t, a_t) \leftarrow r_t + \max_a \hat{Q}(s_{t+1}, a)$$

1. observe s_t
2. estimate $Q(s_t, a_i)$ ∀i using NN
3. choose min/max a_t
4. play a_t
5. estimate $Q(s_{t+1}, a_i)$ ∀i using NN
6. choose max/min a_{t+1}
7. back propagate $\Delta \mathbf{W}$ = (reward + $Q(s_{t+1}, a_{t+1})$)

Neural Network as an Opponent in Quoridor

To speed up the process, I have created 'Heuristic' player following shortest path to goal or placing wall on the shortest path of the opponent.

Neural Network estimating Q-values has been used when learning othello, chess or go.

www.ai.rug.nl/~mwiering/GROUP/ARTICLES/paper-othello.pdf
www.ai.rug.nl/~mwiering/GROUP/ARTICLES/learning-chess.pdf
www2.kobe-u.ac.jp/~ozawasei/pub/iconip02a.pdf

Neural Network as an Opponent in Quoridor

- State

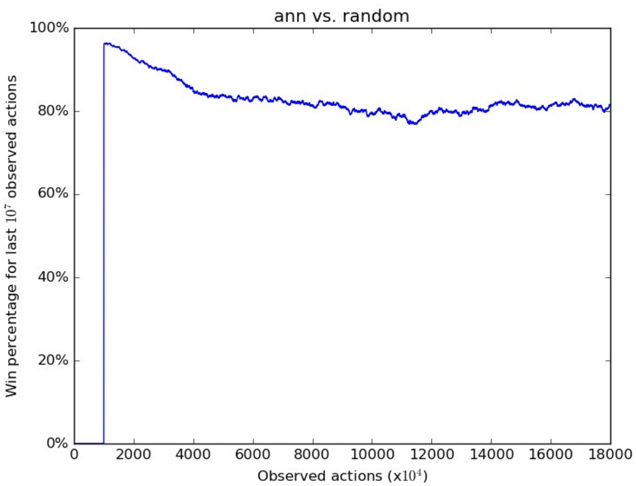
$$S=(c, p_y, p_g, s_y, s_g, w)$$

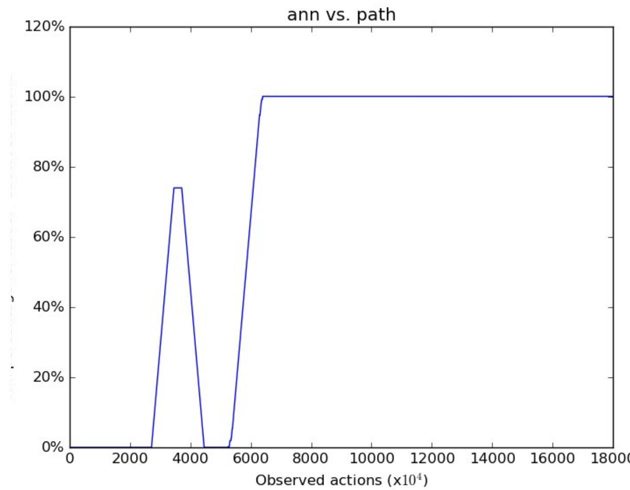
- Context

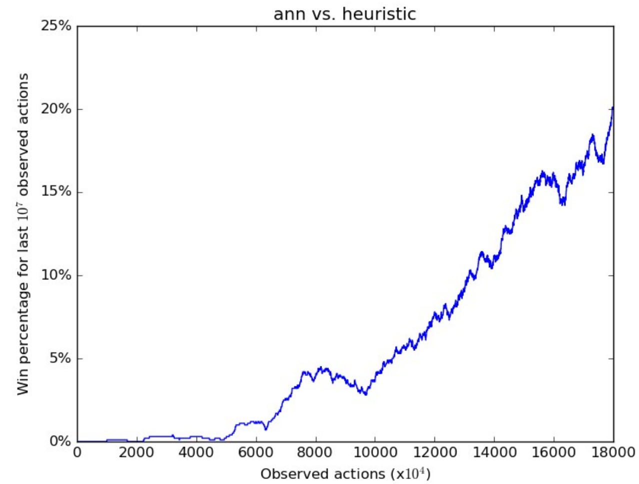
  - shortest path, blockers, crossers

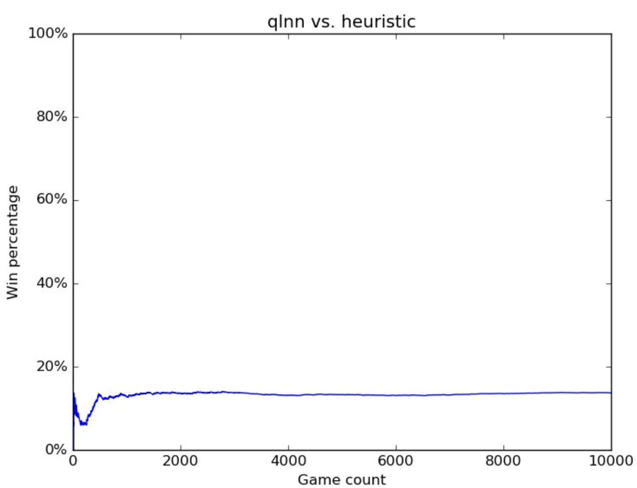
- Tensorflow – python GIL, GPU

Neural Network as an Opponent in Quoridor

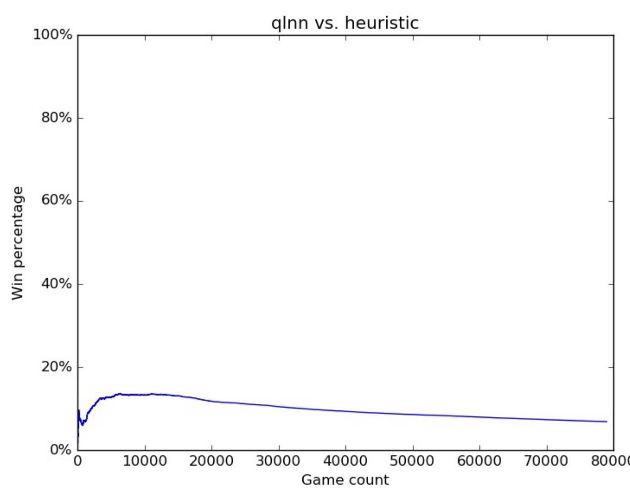




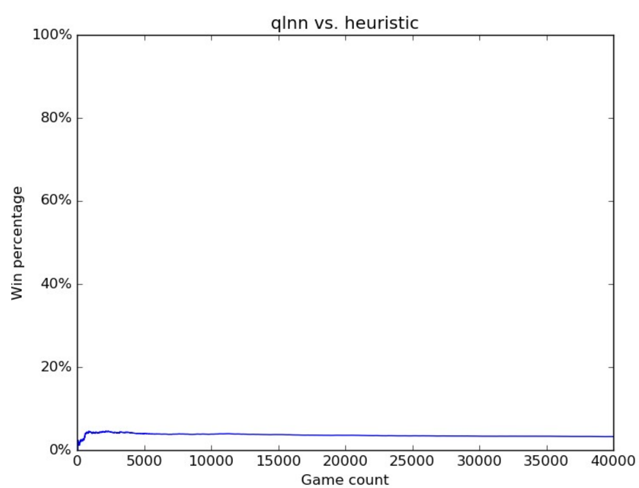




(133, 1000, 140)



(133, 500, 140)



(133, 300, 300, 140)