Artificial Neural Network For The Assessment Of Probability Of Winning From A Game State In Dobo

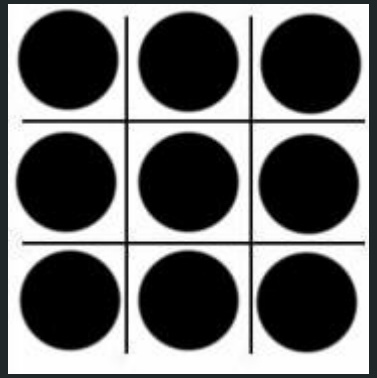
Jake Sauter

General Project Tasks

- Model the game
- Model a Game State table
- Train the game state table
- Model a forward propagating neural neural network
- Train the neural network using a genetic algorithm

Modelling The Game

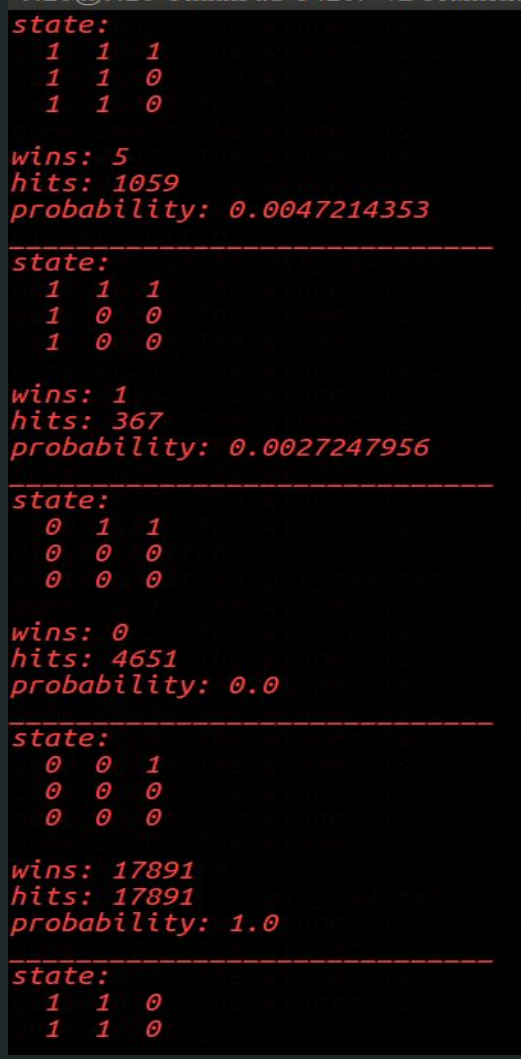
Dobo is a game played on an m x n dimensional board in which a valid move consists of removing any horizontal or vertical contiguous line of stones. The goal of the game is to make your opponent pick up the last stone.



When I initially modelled the game in lisp I used a list to represent the board, however I found that a 2-dimensional array was a better representation and remodelled the game using arrays instead.

Modelling The Game State Table

A game state table was modeled in which each entry was a state-probability tuple, containing a representation of a board and the stored probability of winning from the state. A similarity predicate was implemented to map up to 7 states down to 1 using symmetry to help the convergence of the probability values.



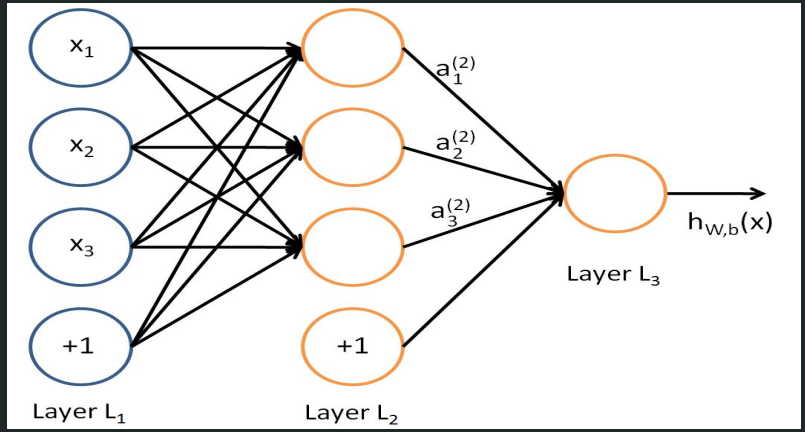
Training the Game State Table

A player who used the game state table as its knowledge base played against a random player, and any time the random player won, the game state table was updated by analyzing the game, increasing the probability of winning from states in the winning sequence and decreasing the probability of winning for states in the losing sequence.

Modelling A Forward Propagating Neural Network

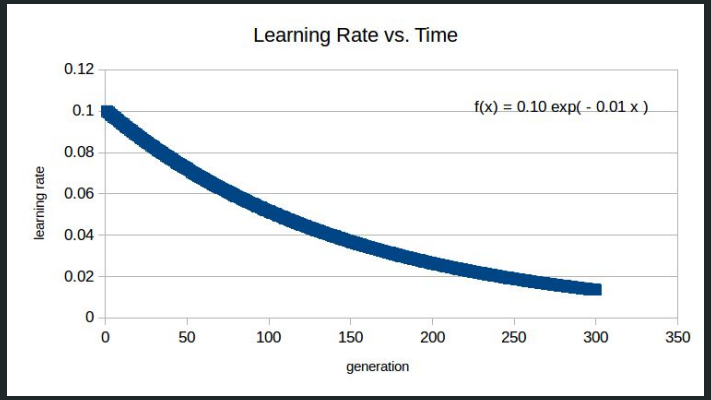
A neural network is nothing more than a system composed of elementwise multiplications, matrix multiplications, matrix additions, and mostly nonlinear activation functions. I've provided a more precise interpretation for a 3 layer neural network.

H(x) = Sigmoid(weights[1] * Sigmoid(x * weights[0] + biases[0]) + biases[1])



Training The Neural Network

I used a genetic algorithm approach to train the neural network. I tried lots of different configurations to find the best way to train it, however there were too many variables for it to be intuitive.



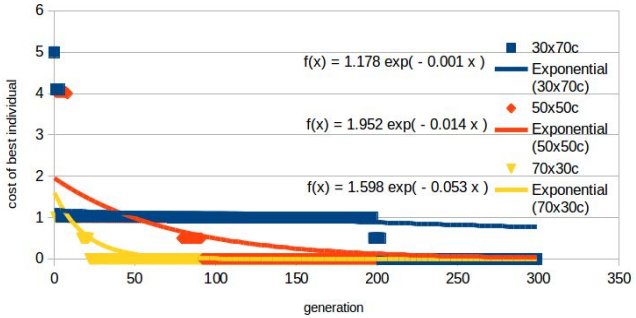
Variables

- Percent Mutation
- Percent Crossover
- Percent Copies
- Mutation Aggression

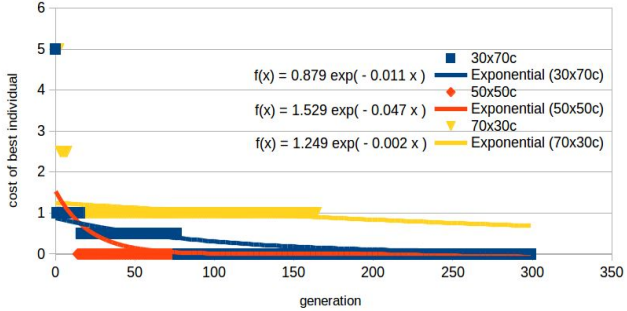
 With an asymptotically decreasing learning rate

so I gathered some data . . .

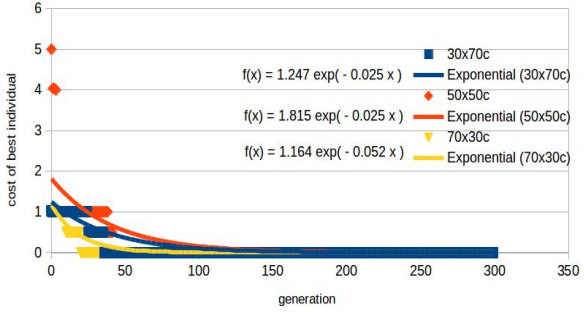
30% mutation, 1/10 mutation aggression



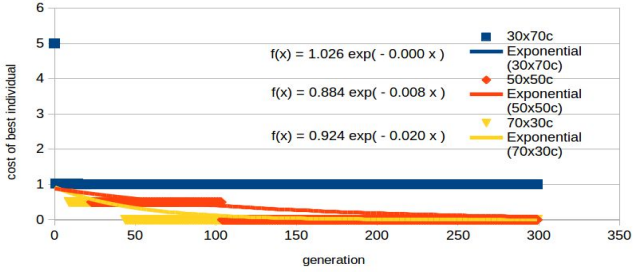
50% mutation, 1/10 mutation aggression



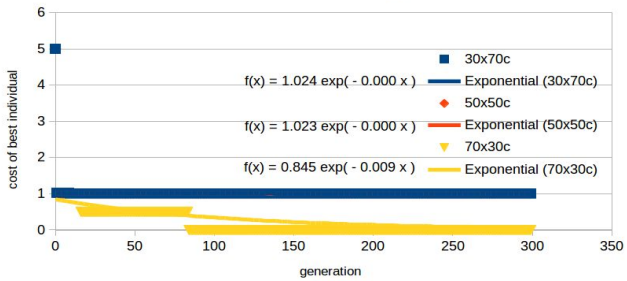
70% mutation, 1/10 mutation aggression



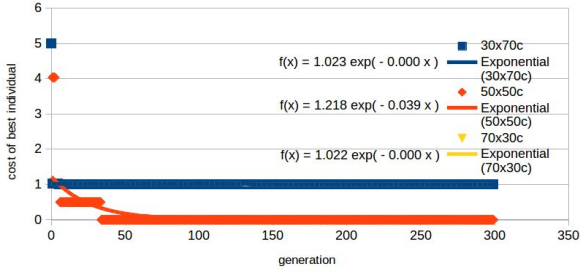
30% mutation, 1/5 mutation aggression



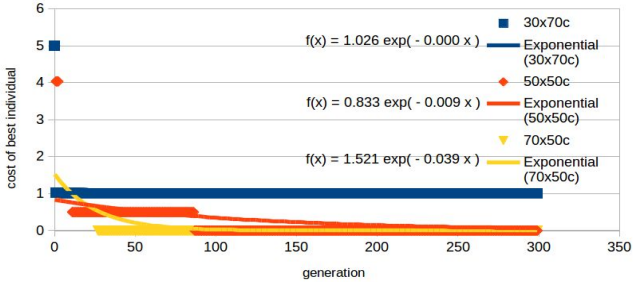
50% mutation 1/5 mutation aggression



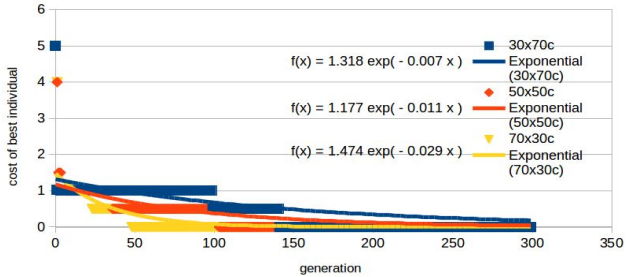
70% mutation, 1/5 mutation aggression



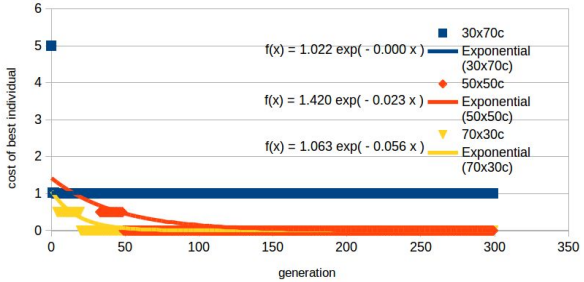
30% mutation 1/3 mutation aggression



50% mutation 1/3 mutation aggression



70% mutation 1/3 mutation aggression

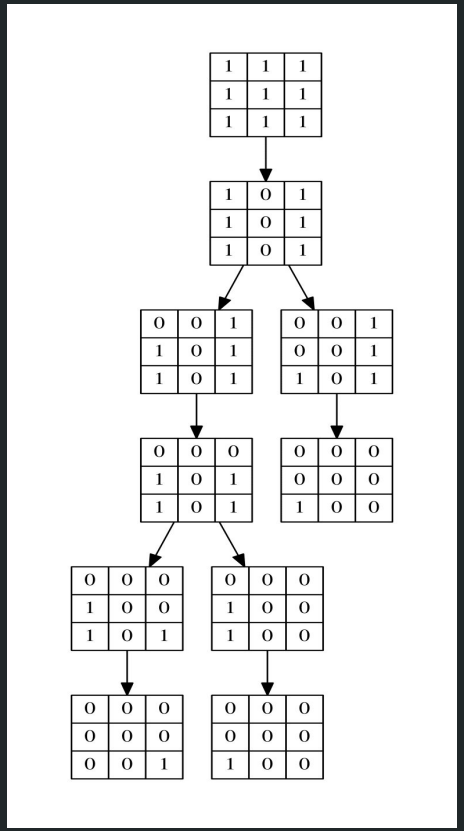


Success?


Writing gst to gst_4x4.txt
game 3281: The winner is . . . TABLE-LOOKUP-PLAYER
game 3282: The winner is . . . RANDOM-MACHINE
game 3283: The winner is . . . TABLE-LOOKUP-PLAYER
game 3284: The winner is . . . TABLE-LOOKUP-PLAYER
game 3285: The winner is . . . RANDOM-MACHINE
game 3286: The winner is . . . TABLE-LOOKUP-PLAYER
game 3287: The winner is . . . TABLE-LOOKUP-PLAYER
game 3288: The winner is . . . TABLE-LOOKUP-PLAYER
game 3289: The winner is . . . TABLE-LOOKUP-PLAYER
game 3290: The winner is . . . TABLE-LOOKUP-PLAYER
Player Switch
Writing gst to gst_4x4.txt
game 3291: The winner is . . . TABLE-LOOKUP-PLAYER
game 3292: The winner is . . . TABLE-LOOKUP-PLAYER
game 3293: The winner is . . . TABLE-LOOKUP-PLAYER
game 3294: The winner is . . . TABLE-LOOKUP-PLAYER
game 3295: The winner is . . . TABLE-LOOKUP-PLAYER
game 3296: The winner is . . . TABLE-LOOKUP-PLAYER
game 3297: The winner is . . . TABLE-LOOKUP-PLAYER
game 3298: The winner is . . . TABLE-LOOKUP-PLAYER
game 3299: The winner is . . . RANDOM-MACHINE
game 3300: The winner is . . . TABLE-LOOKUP-PLAYER
Player Switch
Writing gst to gst_4x4.txt
game 3301: The winner is . . . TABLE-LOOKUP-PLAYER
game 3302: _
Writing gst to gst_5x5.txt
game 991: ^[The winner is . . . TABLE-LOOKUP-PLAYER
game 992: The winner is . . . TABLE-LOOKUP-PLAYER
game 993: The winner is . . . TABLE-LOOKUP-PLAYER
game 994: The winner is . . . TABLE-LOOKUP-PLAYER
game 995: The winner is . . . TABLE-LOOKUP-PLAYER
game 996: The winner is . . . TABLE-LOOKUP-PLAYER
game 997: The winner is . . . TABLE-LOOKUP-PLAYER
game 998: The winner is . . . TABLE-LOOKUP-PLAYER
game 999: The winner is . . . TABLE-LOOKUP-PLAYER
game 1000: The winner is . . . RANDOM-MACHINE
Player Switch
Writing gst to gst_5x5.txt
game 1001: The winner is . . . TABLE-LOOKUP-PLAYER
game 1002: The winner is . . . TABLE-LOOKUP-PLAYER
game 1003: The winner is . . . TABLE-LOOKUP-PLAYER
game 1004: The winner is . . . TABLE-LOOKUP-PLAYER
game 1005: The winner is . . . TABLE-LOOKUP-PLAYER
game 1006: The winner is . . . TABLE-LOOKUP-PLAYER
game 1007: The winner is . . . TABLE-LOOKUP-PLAYER
game 1008: The winner is . . . TABLE-LOOKUP-PLAYER
game 1009: The winner is . . . TABLE-LOOKUP-PLAYER
game 1010: The winner is . . . TABLE-LOOKUP-PLAYER
Player Switch
Writing gst to gst_5x5.txt
game 1011: _
output: 0
actual: 0.0
cost : 0.0
0 0 0
1 0 0
0 0 0
output: 0
actual: 1.0
cost : 1.0
0 0 0
0 0 0
0 0 0
output: 0
actual: 0.0
cost : 0.0
Bad entries: 30.0
retention rate: 0.7777778%
NIL
[9]> _

Reflections/Updates

- I believe that a 9-135-1 configuration would classify every state (and might provide insight on the actual probability value of the state) for the 3x3 case, however I did not have enough time to train it (if any updates are made they will be posted to my website)
- My table player found a perfect game for player 1



Whats Next?

- Implementing Backpropagation
- Adding noise to training
- I will use this project as my basis for understanding of neural networks, and update it as I learn more
- Adding a Convolutional Neural Network Framework for image processing
- An effort to implement the Adam Optimization Technique

Resources

[1] Harris, David J. "Danger of Setting All Initial Weights to Zero in Backpropagation." *Neural Networks - Danger of Setting All Initial Weights to Zero in Backpropagation - Cross Validated*. Stack Exchange, 26 Apr. 2012. Web. 30 Apr. 2017

[2] Michael A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015

[3] Miller, Steven. "Steven Miller." *Mind: How to Build a Neural Network (Part One)*. N.p., 10 Aug. 2015. Web. 30 Apr. 2017.

[4] Montana, David J., and Lawrence Davis. *Training Feedforward Neural Networks Using Genetic Algorithms*. Tech. N.p.: n.p., n.d. Print

[5] Rohrer, Brandon. "How Deep Neural Networks Work." *YouTube*. YouTube, 02 Mar. 2017. Web. 5 Apr. 2017.

[6] Shilov, Georgi E. (1977), Silverman, Richard A., ed., Linear Algebra, Dover, ISBN 0-486-63518

[7] Siegel, Sebastian. *Training an Artificial Neural Network to Play Tic-tac-toe*. Rep. N.p., 20 Dec. 2001. Web. 9 Feb. 2017.

[8] Whitely, D. *Genetic Algorithms and Neural Networks*. Tech. N.p.: n.p., n.d. Print.