Developing an AI Framework to Play Games Without Knowing the Rules

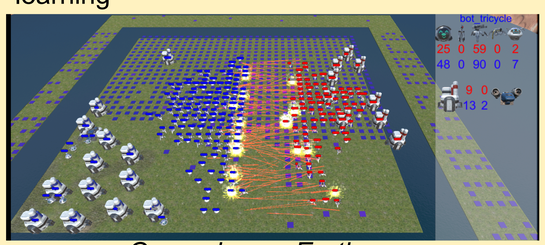
Hieu Nguyen[1] and Ian Schwartz[2]
Faculty Mentor: Dr. Joseph Anderson[2]
[1]Truman College, [2]Salisbury University

Abstract

Game development has improved significantly in decades, especially with AI in games. However, the AI is not sophisticated enough to have a major increase in player's experience. With the new developments in hardware (high-performance computers) and software (parallel programming), we have a chance to improve the AI. The issue is training a machine learning algorithm to play and win games and be able to accomplish this goal in a shorter amount of time by parallelizing the process. In this project, we are attempting to solve the issue by implementing a machine learning algorithm that will play Battlecode games. We approach the problem with two methods: text and image analysis. To preprocess the data, the text method use game logs after the games finished and the image method collect the data while the games are running. We changed parameters of the neural network to see how they affect the performance of the neural network.

Battlecode

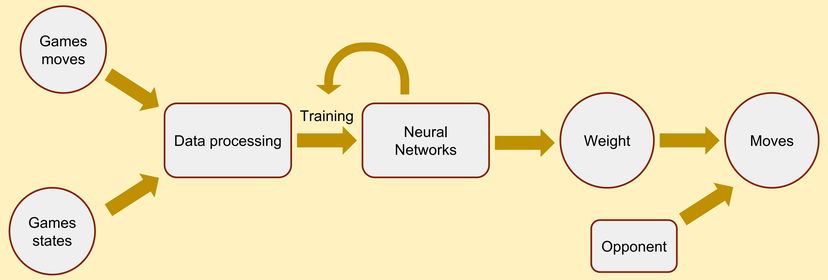
- MIT programming competition
- Real-Time Strategy(RTS) game
- Specifically designed for AI testing
- Allow us to learn about AI and machine learning



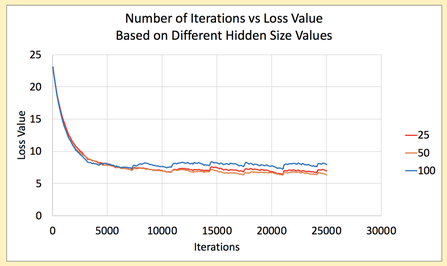
Gameplay on Earth map

Acknowledgement
Supported by NSF (CNS-1757017) Grant to the REU EXERCISE - Explore Emerging Computing in Science and Engineering Program

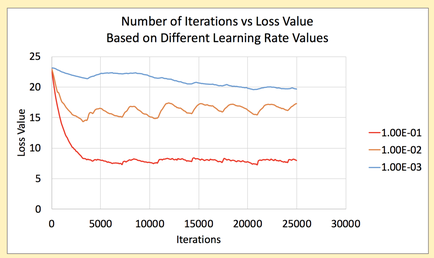
Text Method



Tic Tac Toe

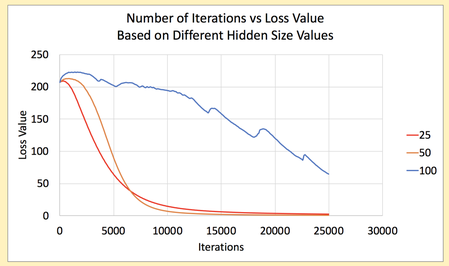


This graph shows how differing the hidden size affects the loss value in Tic Tac Toe

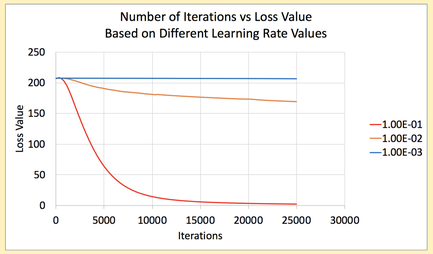


This graph shows how differing the learning rate affects the loss value in Tic Tac Toe

Battlecode



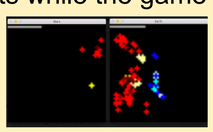
This graph shows how differing the hidden size affects the loss value in Battlecode



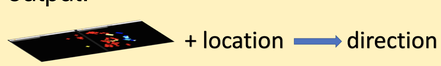
This graph shows how differing the learning rate affects the loss value in Battlecode

Image Method

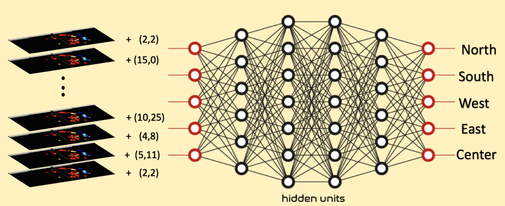
- Record the locations of all units of expert bots while the game is running



- Data, which has RBG images, location of each unit as input and moves as output.



- Data will be used will be used to train the convolutional neural network.



Preliminary results

- 100 game logs recorded and 500 visual data logs
- Figured out how to parse the data (JSON object)
- Wrote scripts in Python that helped us understand Python better (to run the games, to learn how to parse the games, and sorting by who won)
- Ran Karpathy's character neural network on a Shakespeare and Battlecode text file
- Modify the bot code so it can print out the image of each move
- Create a simple battlecode bot that may play the game with trained model later

Future Work

- Have the AI play Tic Tac Toe against another bot
- Take the framework and apply it to much more difficult games (Checkers, Chess, Battlecode)
- Image method:
  - train the data with convolutional neural network,
  - test the model with each type of units,
  - scale up the data with more decisions.

References

Andrej Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks. (2015) Retrieved from http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

MIT programming competition, Battlecode. Retrieved from https://www.battlecode.org (2018)