In this assignment, you will create a replica of Google’s AlphaGo AI to play Go, an abstract strategy board game invented in China over 2,500 years ago. The aim of Go is to surround more territory than the opponent. Between 2015 and 2017, AlphaGo defeated the highest-ranked professional Go players in the world. You will now do the same.

This project is due on December 10, 2018 at 5pm. Starting early is the key to creating an accurate and powerful replica. Because the deadline is very close to the end of the semester, we will not accept any late submissions.

1 Your Assignment

AlphaGo uses a Monte Carlo tree search algorithm to find its moves based on knowledge it has previously learned from an artificial neural network. You will first create an artificial neural network (ANN), train it, and then you will implement a Monte Carlo tree search algorithm to find previous moves.

To ensure your interface is up to our standards, starter code will be provided to you.

Your ANN will be created using neurons. Functionality for your neuron will be implemented through a class that you can create, but make sure it implements the Neuron interface.

The training module will be implemented in a class that implements the Training interface.

Finally, your program will be fed in an AlphaGo game and you are responsible for supplying moves to the game engine through a brain. The interface you will implement for this is called GoGame. Our game engine will be another version of AlphaGo, which you will need to defeat in order to pass this assignment.

2 Details

The game of Go, like many classic games, has a few variations. We will be using the rules officially prescribed by the American Go Association, which may be found at http://www.usgo.org/files/pdf/conciserules.pdf. You may want to view a few tutorials and play Go with a classmate to familiarize yourself with the gameplay.

You can find specifications of the AlphaGo program once you have signed a non-disclosure agreement with Google. Information to sign this agreement and to access the documents can be found on Piazza. You will not be provided the source code of AlphaGo. Instead, the specifications should be enough to allow you to implement your own version.

It is up to you to find sources of information to read in order to create an ANN and to train it. A simple explanation of Monte Carlo algorithms can be found on Wikipedia.

In order to successfully complete this assignment, your AlphaGo implementation should not only replicate the original AI, but also be modified to defeat Google’s implementation. For reference, we’ll be using the AlphaGo Zero algorithm as our benchmark. Google published a paper on this algorithm, which we recommend you read: https://www.nature.com/articles/nature24270

In order to run and train your artificial neural network, we will provide you with access to UT Austin’s supercomputer, Stampede2. More information on access to Stampede2 is specified on our course’s Piazza page. For more information on what Stampede2 is, see this page: https://www.tacc.utexas.edu/systems/stampede2

3 Karma

After Google created AlphaGo Zero, it generalized the program into a super efficient general algorithm called AlphaZero. Generalize your algorithm so that it can be used to not only play Go, but other popular games such as chess. You can even apply your AlphaZero derivative algorithm to play Tetris. See if you can create a better Tetris brain than your original submission through AlphaZero.
4 Submission

Since your project’s submission will be comprised of several files, all files will need to be contained in a ZIP file called prog8.zip. Your submission is due on December 10, 2018 at 5pm on Canvas.

As always, you will need to write a report and turn it in with your submission. Follow the format provided on the previous assignments, and make adjustments for this assignment as you believe are appropriate.

5 Acknowledgements

This assignment was created by Calvin Lin with assistance from Matt Alden, Walter Chang, and Mike Scott.

We thank Google for providing the AlphaGo source code to us and the specifications to students of this class, making this assignment possible.

We also thank TACC for allowing students of this class to use the Stampede2 supercomputer, so that students may use a computer capable of handling artificial neural networks on a large scale such as AlphaGo’s.