Playing Othello by Deep Learning Neural Network

Supervisor: Prof. Kwok-Ping Chan

Members: Nian Xiaodong (3035087112)
         Sun Peigen (3035084548)
         Xu Chaoyi (3035084328)

Introduction

Just half a year ago, AlphaGo made its own fame by a victory against the professional Go player, Mr. Lee Se-dol. And in our project, our objective is also to develop a similar program, applying the same technologies as AlphaGo to play a similar game. The main technique used by AlphaGo is deep learning neural network. As a field which have developed astonishingly these years, deep learning benefits from the huge improvement of computational capability of modern processors and becomes the most popular research topic of artificial intelligence. The game chosen in our project is Othello for simplification. The reason is that averagely, the total number of steps and the number of possible moves during one step are both much smaller than Go (since the board is smaller, and there are limited positions for putting new disks. For the formal definition of Othello, please refer to the appendix).

Methodology

We will divide our work process into three phases.

In the first phase we will confirm our working environment, and planning our future work flow. The final product will include two parts: the front-end and the back-end. We will use HTML5 and Javascript for front-end programs and Python for back-end ones, and the code management tool will be Git and we will adopt Github to store our codes.

In the second phase, we will make the model and then apply supervised learning to
train it. We have got chess manuals of world championship from worldothello.org as our training dataset. Models with distinctive architectures will be built in our project. Evaluation of these models is based on two major criterias: winning rate and reaction speed. The winning rate declares the major performance of the model while the reaction speed is the supporting indicator. These two objectives, generally, can not be achieved at the same time: a higher winning rate usually comes with a longer reaction time. We will combine these two measurements and pick the most appropriate model.

In the third phase, since a certain degree of proficiency will have been reached, it will be trained further by keep playing against itself, using reinforcement learning. Other alternative opponents can be us, visitors of our project website, or other programs playing Othello on the Internet. This procedure is exactly how AlphaGo is trained.

**Scope**

The languages we are going to use in this project is Python for its powerful packages compared to other languages. Another reason is that all of our group members are familiar with Python, so we can save the time for learning a new language.

There are dozens of machine learning packages for Python and after thorough contemplation, we want to choose Keras as our framework. Keras as a new-risen learning package, is specially designed for deep learning and its main feature is the simplicity to build new models. Using Theano and TensorFlow (Theano is a widely used deep learning package and TensorFlow is a machine learning package developed by Google) as computing backend, the speed of Keras is guaranteed while enabling us not to touch the complex model-build process.

The hardware may be involved in the project is the graphics card. The graphics card can accelerate the training speed of the neural networks at great extent and shorten the development cycle, which can free us from keep waiting results and sparing us more time to redesign models. Currently, the graphic cards we are going to use is Nvidia GT 640, as it is cheap and has sufficient computing ability. The possibility that
we switch to another more powerful graphic cards still can not be ruled out depending on the actual scale of our model.

**Deliverables**

1. A graphic gameboard interface to interact with the program
   
   We decide to design a graphic user interface for the program to interact with human beings instead of output the recipes or strings. The gameboard graphic interface should be able to capture human players’ movement on the checkerboard, present the program’s response and reverse the pieces based on the rules of Othello. This interface will serve as the game engine and calculate the change of stones after one’s move.

2. A policy network from supervised learning
   
   A policy network built from supervised learning using the probability distribution from the rollout policy is necessary to predict moves of other top human players and AI players. The neural network will be an evaluation function to estimate the strength of each move and help to give the basic choice of each move for the computer.

3. A policy network from reinforcement learning
   
   A policy network is required to so self-playing and keep reinforce the program by avoiding overfitting. We will use this network to perform self-playing for thousands of games and help to re-tune the weights (parameters) to improve the winning rate of the model.
### Time Schedule & Milestones

<table>
<thead>
<tr>
<th>Stage</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completion of the webpage and the GUI of Othello construction</td>
<td>Sep-30-2016</td>
</tr>
<tr>
<td>Gathering the training data we need later</td>
<td>Oct-15-2016</td>
</tr>
<tr>
<td>Train the supervised policy network and choose the appropriate model</td>
<td>Nov-15-2016</td>
</tr>
<tr>
<td>Train the reinforcement policy network</td>
<td>Dec-20-2016</td>
</tr>
<tr>
<td>Settle the first release and do the first presentation</td>
<td>Jan-10-2017</td>
</tr>
<tr>
<td>Retune the model and try different structures and algorithms</td>
<td>Mar-30-2017</td>
</tr>
<tr>
<td>Complete the final release and submit the final report</td>
<td>Apr-15-2017</td>
</tr>
</tbody>
</table>

### Conclusion

The objective of our project is the application of deep learning on Othello. We expect that the program developed by us can achieve, if not transcend, the level of traditional algorithms based on game-tree search. Hopefully the pipeline of our research can also be adopted in other game AI engines.
Appendix

Definition of Othello: Reversi is a strategy board game for two players, played on an 8×8 uncheckered board. There are sixty-four identical game pieces called disks (often spelled "discs"), which are light on one side and dark on the other. Players take turns placing disks on the board with their assigned color facing up. During a play, any disks of the opponent's color that are in a straight line and bounded by the disk just placed and another disk of the current player's color are turned over to the current player's color. The object of the game is to have the majority of disks turned to display your color when the last playable empty square is filled. (https://en.wikipedia.org/wiki/Reversi)