UCT-Enhanced Deep Convolutional Neural Networks For Move Recommendation in Go

Sarun Paisarnsrisomsuk
Pitchaya Wiratchotisatian
Go: An Ultimate Challenge for AI

- An ancient board game
- Two-player
- Zero-sum
- Deterministic
- Perfect-information
Simple Rules, but Complex Strategies

- Place stones in turn on a 19x19 board
- Basic goal: secure more territories than the opponent
- Enormous combinatorial complexity
- Long-term influence of a move
Goal

To enhance move recommendation in Go using DCNN and UCT
Deep Convolutional Neural Networks (deep CNN)
Neural Network

http://cs231n.github.io/neural-networks-1
Neural Network

http://en.wikipedia.org/wiki/Artificial_neural_network
Deep Convolutional Neural Networks

http://deeplearning.net/tutorial/lenet.html
Deep Convolutional Neural Networks

Upper Confident Bounds applied to tree (UCT)

Applied bandit-based method to guide Monte-Carlo planning
Monte Carlo Tree Search

[Diagram showing the process of Monte Carlo Tree Search]

http://ccg.doc.gold.ac.uk/papers/browne_tciaig12_1.pdf
Upper Confident Bounds applied to tree (UCT)

Selection Policy:

\[ UCT = \bar{X}_j + 2C_p \sqrt{\frac{2 \ln n}{n_j}} \]

Choose node which has maximum value of UCT
Methodology
Go Data

- 170,000 complete games from the KGS Server (Kiseido Go database)
- Extracted features from each individual move

<table>
<thead>
<tr>
<th>Feature</th>
<th>Planes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black / white / empty</td>
<td>3</td>
<td>Stone colour</td>
</tr>
<tr>
<td>Liberties</td>
<td>4</td>
<td>Number of liberties (empty adjacent points)</td>
</tr>
<tr>
<td>Liberties after move</td>
<td>6</td>
<td>Number of liberties after this move is played</td>
</tr>
<tr>
<td>Legality</td>
<td>1</td>
<td>Whether point is legal for current player</td>
</tr>
<tr>
<td>Turns since</td>
<td>5</td>
<td>How many turns since a move was played</td>
</tr>
<tr>
<td>Capture size</td>
<td>7</td>
<td>How many opponent stones would be captured</td>
</tr>
<tr>
<td>Ladder move</td>
<td>1</td>
<td>Whether a move at this point is a successful ladder capture</td>
</tr>
<tr>
<td>KGS rank</td>
<td>9</td>
<td>Rank of current player</td>
</tr>
</tbody>
</table>
New Feature

- Board pattern at the end of game, or "final board pattern"
Deep Convolutional Neural Network

- We implemented a small deep CNN
  - 1 hidden layer; no pooling
  - 10 kernels
Result from Adding Final Board Pattern to input

- Without final board pattern, accuracy 6%
- With final board pattern, accuracy 18%
- The actual final board pattern is not possible to be obtained
UCT-Simulated Final Board Pattern

- Collect final board pattern in each simulation during UCT
Experiments and Results
Result
Result
Conclusion and Future Work
Conclusion

- Adding final board pattern to inputs of the deep CNN improves the accuracy of the network.
- In practice, we can collect statistics in each simulation of UCT to approximate final board pattern.
Future Work

- Deep network’s size
- More Training Data
- Additional Features
- Combining deep network with UCT
Acknowledgements

- Levente Kocsis, Project Advisor and SZTAKI liaison
- Gabor Sarkozy, MQP Advisor
- Worcester Polytechnic Institute
- MTA-SZTAKI
- SZTAKI Colleagues
- Pachi and Gnu Go Development Team
Questions?
Köszönöm szépen!

Köszönöm

Köszönöm szépen!