Intelligent Non-Player Character with Deep Learning

Meng Zhixiang, Zhang Haoze
Supervised by Prof. Michael Lyu
CUHK CSE FYP Term 1
Intelligent Non-Player Character with Deep Learning
Background

We all know the results...
Agenda

- Background
- Motivation & Objective
- Methodologies
- Design & Implementation
- Results & Discussion
- Conclusion
Agenda

- Background
  - Development of AI in Go, Chess and Chinese Chess
  - Difference among Go, Chess and Chinese Chess
- Motivation & Objective
- Methodologies
- Design & Implementation
- Results & Discussion
- Conclusion
Development of AI in Go

No Good Results

Zen beat Takemiya Masaki at five stones handicap Mar 2012

AlphaGo beat Lee Sedol Mar 2016

Minimax Searching Pruning

Monte Carlo

Deep Learning

Intelligent Non-Player Character with Deep Learning
Difference between Go and Chess/Chinese Chess
Development of AI in Chess

- **Deep Blue** beat Garry Kasparov May 1997
- **Stockfish** won TCEC 2013, 2014, 2015
- **Giraffe** plays at the level of an FIDE International Master on a PC Sep 2015

- Minimax Searching Evaluation Function Hand-Coded Knowledge
- Deep Reinforcement Learning

**TCEC**: Top Chess Engine Championship  
**FIDE**: World Chess Federation
Difference between Chess and Chinese Chess
Development of AI in Chinese Chess

Tiansuo Inspur System beat five Grandmaster players Aug 2006
Minimax Searching Alpha-Beta Pruning Hand-Coded Knowledge

Chess Nade beat three Master players Nov 2009
Minimax Searching Alpha-Beta Pruning Hand-Coded Knowledge

Now ???
Deep Learning ???

Intelligent Non-Player Character with Deep Learning
Motivation
Objective

Human Player -> User Interface -> Server

Server -> Game AI -> User Interface

Intelligent Non-Player Character with Deep Learning
Agenda

- Background
- Motivation & Objective
- Methodologies
  - Supervised Learning
  - Convolutional Neural Network
- Design & Implementation
- Results & Discussion
- Conclusion
Supervised Learning

- **Supervised Learning**
  - the “right answer” is given
  - Regression Problem & Classification Problem

- **Unsupervised Learning**
  - no “right answer” is given
  - Clustering Problem
Neural Network

- Non-linear Hypotheses
- Neurons and Brain
- Backpropagation
Convolutional Neural Network

- Feed-forward
- Organization of Animal Visual Cortex
- Image Recognition
Agenda

- Background
- Motivation & Objective
- Methodologies
  - Design & Implementation
    - Project Workflow
- Results & Discussion
- Conclusion
Project Workflow

Model Design → Model Building → Model Training → Model Testing

Accuracy Testing → Real Performance Testing
Design Overview

Game AI

Policy Network
Predict probabilities of next moves

Evaluation Network
Evaluate winning rate
Game AI Structure

Message Receiver → Format Converter → Feature Extractor → Piece Selector → Decision Maker → Message Sender

- Move Selector
## Feature Channels

<table>
<thead>
<tr>
<th>Feature Channel 1</th>
<th>Pieces belonging to different sides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Channel 2</td>
<td>Pieces of Advisor type</td>
</tr>
<tr>
<td>Feature Channel 3</td>
<td>Pieces of Bishop type</td>
</tr>
<tr>
<td>Feature Channel 4</td>
<td>Pieces of Cannon type</td>
</tr>
<tr>
<td>Feature Channel 5</td>
<td>Pieces of King type</td>
</tr>
<tr>
<td>Feature Channel 6</td>
<td>Pieces of Knight type</td>
</tr>
<tr>
<td>Feature Channel 7</td>
<td>Pieces of Pawn type</td>
</tr>
<tr>
<td>Feature Channel 8</td>
<td>Pieces of Rock type</td>
</tr>
<tr>
<td>Feature Channel 9</td>
<td>Valid moves for the selected piece</td>
</tr>
<tr>
<td>(only for Move Selector)</td>
<td></td>
</tr>
</tbody>
</table>
Feature Channels

Intelligent Non-Player Character with Deep Learning
Piece Selector & Move Selector

<table>
<thead>
<tr>
<th></th>
<th>r</th>
<th>n</th>
<th>b</th>
<th>a</th>
<th>k</th>
<th>a</th>
<th>b</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>c</td>
<td>n</td>
<td>c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>C</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>N</td>
<td>B</td>
<td>A</td>
<td>K</td>
<td>A</td>
<td>B</td>
<td>N</td>
<td>R</td>
</tr>
</tbody>
</table>

Intelligent Non-Player Character with Deep Learning
Piece Selector & Move Selector

- Extracted Features
- First Hidden Convolutional Layer
- Second Hidden Convolutional Layer
- Third Hidden Layer (Softmax Layer)
- Probability Distribution

Rectified Linear Unit (ReLU)
Selection Strategy

- **Strategy 1:**
  - Select the piece with highest possibility given by Piece Selector
  - Select the destination of that piece with highest possibility given by Move Selector

- **Strategy 2:**
  - Calculate the probability of moving a piece * the probability of a destination of that piece
  - Select the combination with highest probability
Project Workflow

Model Design → **Model Building** → Model Training → Model Testing

Accuracy Testing → Real Performance Testing
TensorFlow

- an open source software library
- for numerical computation
- using data flow graphs
- flexibility and portability
Project Workflow

Model Design → Model Building → **Model Training** → Model Testing →

- Accuracy Testing
- Real Performance Testing
Training Dataset

Collected Game Records

Features and Targets

Training Dataset for Different NN models
FEN Format

[rnbakab1r/111111111/1c1111nc1/p1p1p1p1p/111111111/111111111/RNBAKABNR, r]
Format Conversion

炮二平五  炮8平5
马二进三  马8进7
车一进一  车9平8
车一平六  车8进6
车六进七  马2进1
车九进一  炮2进7
炮八进五  马7退8
炮五进四  士6进5
车九平六  将5平6
前车进一  士5退4
车六平四  炮5平6
车四进六  将6平5
炮八平五
Training Strategy

- Piece Selector and Move Selector are trained separately
- Shuffle the training dataset containing over 1,600,000 moves
- Train the models batch by batch
- Test the accuracy along the process
- An untrained testing dataset containing over 80,000 moves
Project Workflow

Model Design → Model Building → Model Training → Model Testing

Accuracy Testing → Real Performance Testing
Results

Piece Selector Accuracy

accuracy = # of correct predictions / total # of test cases

prediction: the choice with the highest probability
Results

Move Selector Accuracy
## Results

<table>
<thead>
<tr>
<th>Move Selector</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advisor</td>
<td>89.8%</td>
</tr>
<tr>
<td>Bishop</td>
<td>91.2%</td>
</tr>
<tr>
<td>Cannon</td>
<td>54.1%</td>
</tr>
<tr>
<td>King</td>
<td>79.8%</td>
</tr>
<tr>
<td>Knight</td>
<td>70.1%</td>
</tr>
<tr>
<td>Pawn</td>
<td>90.4%</td>
</tr>
<tr>
<td>Rock</td>
<td>53.6%</td>
</tr>
</tbody>
</table>

Move Selector Accuracy
Results
Results
Results
Results

Selection Strategy 1

Selection Strategy 2
Discussion

- Possible Reasons:
  - CNN not deep enough
  - Training dataset not large enough
  - Records in training dataset may not be the optimal choices
    - For one chessboard status, there may be different move choices in training dataset
    - It’s hard to judge which choice is better in current phase
Conclusion

- Achieved overall high accuracy
- Performed badly in some cases
- Need further improvement
  - Reinforcement Learning
    - Not limited by training dataset
  - Evaluation Network
    - To judge which move is better
Q&A