Go - Game

by Yang

17th May 2023

go.py

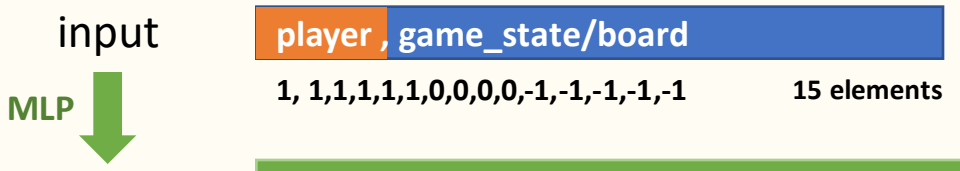
Code for playing 1d go game

- game state = [1,1,1,1,1,0,0,0,0,-1,-1,-1,-1,-1]
- player = - last player
- validate func: check if move is valid
  - ✓ can't move to same pos where you start
  - ✓ can't move outside the board
  - ✓ check if the piece belongs to player
  - ✓ check if the destination is empty
  - ✓ optional: check for suicide move
- capture func: check if pieces are captured by player
  - ✓ find player piece and check for sequence opponent ended by player
- random move func:
  - ✓ create list of all valid moves (suing validate)
  - ✓ choose one randomly
- greedy move func:
  - ✓ create list of all valid moves (using validate)
  - ✓ chose move that kills most pieces of other player
- interactive move:
  - ✓ ask user to input move
  - ✓ validate

Go Game:

- Running game, create training data, training neural network, plotting and playing the game interactively via Jupyter notebook.

nn.py



index 0 = move (0,1) = let's move from board0 to board1

index 1 = move (0,2) –> similar as above

…

index 182 = move(14,13)

neural network architecture (MLP):

- input 15 dim , output 182 dim, prob of each move (softmax)
- 3 hidden layers with 128 units (dense layer, relu)

training

- standard optimizer (adam)
- training data from (greedy) game: player, state, move (each move is represented in a long array [00010000] 182 dim
- output of network compared with move via cross-entropy(how far I am away from the training data – the smaller the better)

play game:

- run network to get probability for each move
- choose move with maximal probability
- optional: validate:
  - ✓ only choose move from valid moves (using validate from go)

implementation:

- tensorflow / keras with MLP = multiple Dense layers