Session 2.4 Artificial Intelligence for Games

Time & Location: 16:00-17:30, Dec. 1, L008

Chair: Prof. Kiminori Matsuzaki (松崎公紀)

(1) AlphaZero for a Non-deterministic Game

Chu-Hsuan Hsueh (National Chiao Tung University), I-Chen Wu (National Chiao Tung University), Jr-Chang Chen (National Taipei University), and Tsan-sheng Hsu (Academia Sinica)

The AlphaZero algorithm, developed by DeepMind, achieved superhuman levels of play in the games of chess, shogi, and Go, by learning without domain-specific knowledge except game rules. This paper investigates whether the algorithm can also learn theoretical values and optimal plays for non-deterministic games. Since the theoretical values of such games are expected win rates, not a simple win, loss or draw, it is worthy investigating the ability of the AlphaZero algorithm to approximate expected win rates of positions. This paper also studies how the algorithm is influenced by a set of hyper-parameters. The tested non-deterministic game is a reduced and solved version of Chinese dark chess (CDC), called 2×4 CDC. The experiments show that the AlphaZero algorithm converges nearly to the theoretical values and the optimal plays in many of the settings of the hyper-parameters. To our knowledge, this is the first research paper that applies the AlphaZero algorithm to non-deterministic games.

(2) Weighted Majority Voting with a Heterogeneous System in the Game of Shogi

Shogo Takeuchi (Kochi University of Technology)

In this paper, we propose a method of the weighted voting with a heterogeneous system in games and propose to assign the strength of engines and win probabilities of positions to the weights for voting. Assigning the strength as the weight will solve the problem of the weaker engines in the majority voting. Additionally, we introduce a sigmoid function for each engine in order to transform an evaluation value to a win probability. By this sigmoid functions, we can compare the win probabilities between the different engines and resolve the problem in the optimistic voting with a heterogeneous system.

We performed the tournaments among the proposed system and the other voting systems against a single engine to compare the strength of the voting systems in shogi. From the experimental results, we showed that the proposed method defeats all other voting systems.

(3) Learning of Evaluation Functions via Self-Play Enhanced by Checkmate Search

Taichi Nakayashiki (University of Tokyo) and Tomoyuki Kaneko (University of Tokyo)

As shown in AlphaGo and AlphaZero, reinforcement learning is effective in learning of evaluation functions (or value networks) in Go, Chess and Shogi. In their training, two procedures are repeated in parallel; self-play with a current evaluation function and improvement of the evaluation function by using game records yielded by recent self-play. Although AlphaGo Zero and AlphaZero have achieved super human performance, it requires enormous computation resources. To alleviate the problem, this paper proposes to incorporate a checkmate solver in self-play. We show that this small enhancement dramatically improves the efficiency in our experiments in Minishogi, via the quality of game records in self-play. It should be noted that our method is still free from human knowledge about a target domain, though the implementation of checkmate solvers is domain dependent.

(4) An Alternative Multitask Training for Evaluation Functions in the Game of Go

Yusaku Mandai (University of Tokyo) and Tomoyuki Kaneko (University of Tokyo)

For the game of Go, Chess, and Shogi (Japanese Chess), deep neural networks (DNNs) have contributed to build accurate evaluation functions and many studies have attempted to create so called the value network which predicts a reward of a given state. A recent study of the value network for the game of Go has shown that a two-headed neural

network with two different objectives can be trained effectively and performs better than a single-headed network. One of the two heads is called a value head and the other, policy head, predicts next moves at a given state. This multitask training makes the network more robust and improves the generalization performance. In this paper we show that a simple discriminator network is an alternative target of the multitask learning. Compared to the existing deep neural network, our proposed network can be designed more easily because of its simple output. Experimental results showed that our discriminative target also makes the learning stable and the evaluation function trained by our method is comparable to the training of existing studies in terms of predicting next moves and playing strength.

(5) Interpreting Neural-Network Players for Game 2048

Kiminori Matsuzaki (Kochi University of Technology) and Madoka Teramura (Kochi University of Technology)

Game 2048 is a stochastic single-player game and development of strong computer players for 2048 has been based on N-tuple networks trained by reinforcement learning. In our previous study, we developed computer players for game 2048 based on convolutional neural networks (CNNs), and showed by experiments that networks with three or more convolution layers performed much better than 2-convolution network. In this study, we analyze the inner working of our CNNs (i.e. white box approach) to identify the reasons of the performance. Our analysis includes visualization of filters in the first layers and backward trace of the networks for some specific game positions. We had several findings on inner working of our CNNs for game 2048.

(6) Empirical Analysis of PUCT Algorithm with Evaluation Functions of Different Quality

Kiminori Matsuzaki (Kochi University of Technology)

Monte-Carlo tree search (MCTS) algorithms play an important role in developing computer players for many games. The performance of MCTS players is often leveraged in combination with offline knowledge, i.e., evaluation functions. In particular, recently AlphaGo and AlphaGo Zero achieved a big success in developing strong computer Go player by combining evaluation functions consisting of deep neural networks with a variant of PUCT (Predictor + UCB applied to trees). The effect of evaluation functions on the strength of MCTS algorithms, however, has not been investigated well, especially in terms of the quality of evaluation functions. In this study, we address this issue and empirically analyze the AlphaGo's PUCT algorithm by using Othello (Reversi) as the target game. We investigate the strength of PUCT players using variants of an existing evaluation function of a champion-level computer player. From intensive experiments, we found that the PUCT algorithm works very well especially with a good evaluation function and that the value function has more importance than the policy function in the PUCT algorithm.