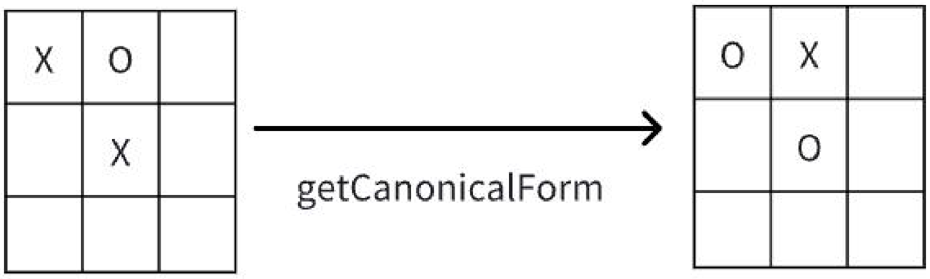
Project4

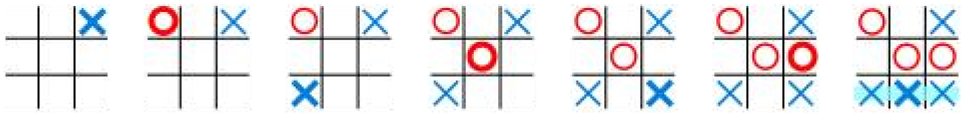
Notice:

In this implementation, we always treat the player to move as player 1, by calling `getCanonicalForm` to reverse the board state. Taking TicTacToe game as an example as below, the player always is `X`, so that the algorithm can easily be applied to both players.



1. MCTS(Monte Carlo Tree Search)

The **TicTacToe** is a game for two players who take turns marking the spaces in a 3x3 grid with X or O. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row first is the winner.



The state of the game is represented by a 3x3 ndarray. The agent has 9 actions in total for every grid in the board, the action is valid if and only if the grid has not been taken.

The reward is 1 for winner, -1 for loser. If ties, the reward is a slightly positive value.

Monte Carlo Tree Search (MCTS) is a heuristic search algorithm widely used in decision-making processes, particularly in games and complex environments. It combines random sampling with tree search to efficiently explore possible actions. MCTS builds a tree by iteratively selecting promising nodes (selection), expanding the tree (expansion), simulating outcomes (simulation), and updating node statistics (backup). This process balances exploration and exploitation, allowing it to focus on high-reward paths while maintaining adaptability. MCTS is especially effective in scenarios with large state spaces, such as Go or chess, where traditional search methods are computationally expensive.

Q1: Implement selection, expansion, simulation, backup stages and the whole roll-out process of MCTS algorithm.
Fill out the """Your code here""" of PureMCTS class in alpha_zero/MCTS.py

2. AlphaZero

Othello is a two player board game, usually played on an 8x8 board, while in this project we use 6x6 board. Players take turns placing one disk on an empty square, with their assigned color facing up. After a play is made, any disks of the opponent's color that lie in a straight line bounded by the one just played and another one in the current player's color are turned over. When all playable empty squares are filled, the player with more disks showing in their own color wins the game.

The state of the game is represented by a 6x6 ndarray. The agent has 36 actions in total for every grid in the board, the action is valid if and only if the grid has not been taken.

The reward is 1 for winner, -1 for loser. If ties, the reward is a slightly positive value.

AlphaZero is a reinforcement learning algorithm developed by DeepMind that masters games like chess, shogi, and Go through self-play. It combines a deep neural network with

Monte Carlo Tree Search (MCTS) to evaluate positions and select moves. The neural network predicts move probabilities and outcomes, guiding MCTS simulations. Through iterative self-play, AlphaZero improves its policy and value predictions, achieving superhuman performance without human knowledge or pre-existing data. Its ability to learn from scratch and generalize across domains showcases its versatility and power in solving complex decision-making problems.

Q2: Implement simulation and UCB selection process of MCTS algorithm in AlphaZero. Implement the neural network for Othello, and the training stage for AlphaZero.

Simulation and UCB selection used in AlphaZero: Fill out the """Your code here""" of \`MCTS\` class in \`alpha_zero/MCTS.py\`

Training stage: Fill out the """Your code here""" in \`alpha_zero/Coach.py\`