Dependency

This package build upon the 2nd tutorial package:`gomoku_gym`, and the anaconda virtural environment named `torch` from the 1st tutorial.

Make sure that you have built the code of previous tutorials

(**Important**: Since there is some modifications in gomoku_gym in tutorial 2, you need to re-download and re-install the latest version of tutoiral 2 package from blackboard system.)

Install

Open a terminal, make sure that conda enviroment is activated. If not, just type:

```
conda activate torch
```

And install the alphazero package

```
cd <path-to-proj_tut3>
pip install -e .
```

TODO

1. Programming part

1.1 Required Part

Please search the tag `#TODO` in the files:

- `submission3_mcts_alphaZero.py`
- `submission3_policy_value_net_pytorch.py`
- `submission3_train.py`

And code the corresponding part. These code can be concluded into three part

- Program a Network for alpha zero using Pytorch (30% points)
- Program a Monte Carlo Tree Search for alpha zero. (30% points)
- Program the training pipeline (30% points)
- Plot the evaluation curve of the trained model against a pure Monte Carlo Tree Search model. (10% points)

There are instruction comments to guide you to finish the coding, except for the plotting part. You can find them under tag `#TODO`

(**Tips**: The code implements a game with 6 * 6 board and 4 in a row. For this case, we may obtain a reasonably good model within 500~1000 self-play games in about 2 hours on a single PC.)

1.2 Bonus Part

There are bouns part that required you to tweak or program the code.

- Train alpha zero to play a 8x8 board with 5-in-a-row winning. (20% extra points)

(**Tips**: For the case of 8 * 8 board and 5 in a row, it may need 2000~3000 self-play games to get a good model, and it may take about 2 days)

2. Presentation

You are required to prepare presention for the alpha zero demo corresponding to Tutorial #3. The presenting content is flexible. For example, you can introduce implementing details of your code; you can show the video or live demo to play against alpha zero; you can shows the plotting results.

Grading Criteria for Course Project

We have 3 tututorials. Here are the scoring weights for each tutorial:

- Tutorial #1: 10%
- Tutorial #2: 10%
- Tutorial #3: 60%
- Presentation: 20%

Here are an example of calculation:

```
Student A:
    - tutorial #1: 90 points
    - tutorial #2: 100 points
    - tutorial #3:
      - Program a Network for alpha zero using Pytorch (30 points)
      - Program a Monte Carlo Tree Search for alpha zero. (30 points)
      - Program the training pipeline (30 points)
      - Plot the evaluation curve of the trained model against a pure Monte
Carlo Tree Search model.  (10 points)
      - Train alpha zero to play a 8x8 board with 5-in-a-row winning. (20
extra points)
    - Presentation: 18 points


Overall scores = 90x10% + 100x10% + (30+30+30+10)x60% + 18x20%
```