CSC411: Assignment 4 bonus

Due on Wednesday, April 4, 2018

Yu-Chien Chen, Hunter Richards

Problem 1

The starting player's turn was randomly determined using a ”coin flip” (i.e. a randomly generated integer between 1 and 2 inclusive) for each episode. Considering the model is trained for 100000 iterations, the split between player one going first and player two going first is approximately 50/50. Some hyperparameters have been changed since Project 4, such as setting the invalid move return from -1 to -2 and increasing the number of training episodes from 50000 to 100000. The number of hidden units has remained fixed at 96.

The following graph displays the learning curve of the model:

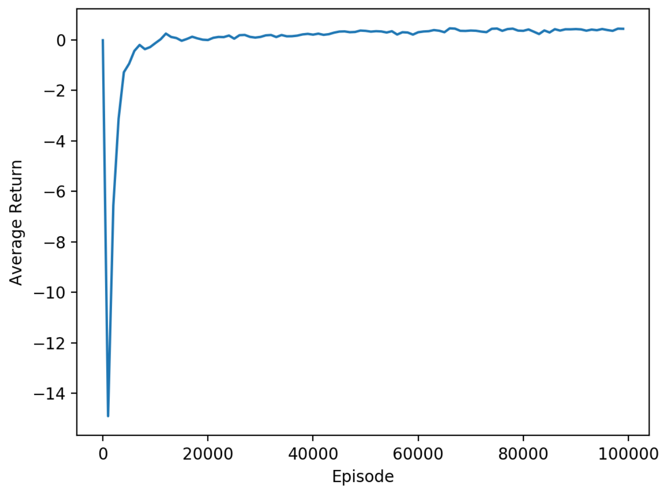

Figure 1: Average Return vs Episode

and the following displays the win rate across episodes:

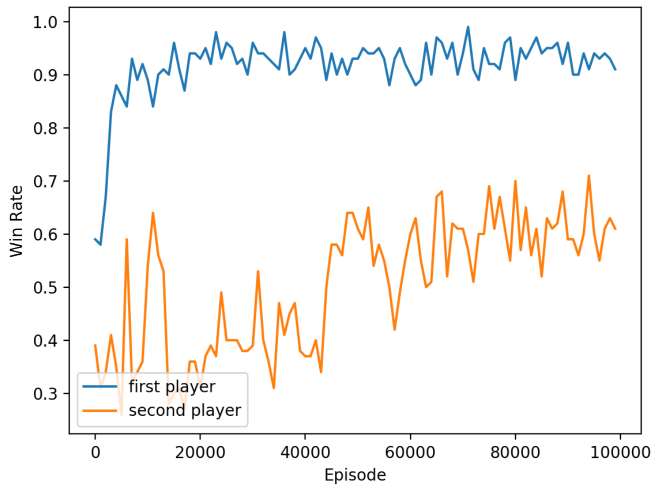

Figure 2: Win Rate vs Episode

We can see that this is a much harder problem than Project 4. The win rate of the model when it plays second is consistently less than the win rate when it plays first. The model can no longer rely on securing a position in a corner of the board from the start, and must learn to play tactically.

```
Game 0      Game 80     Game 20     Game 60     Game 80
...         ...         ...         ...         ...
.x.         .x.         .o.         .o.         .o.
...         ...         ...         ...         ...
====        ====        ====        ====        ====
...         ...         ...         x..         x..
.x.         .x.         .o.         .o.         .o.
.o.         o..         .x.         ...         ...
====        ====        ====        ====        ====
..x         x..         ...         x..         xo.
.x.         .x.         .oo         .o.         .o.
.o.         o..         .x.         ..o         ...
====        ====        ====        ====        ====
..x         x..         ...         x.x         xox
.xo         .x.         .oo         .o.         .o.
.o.         oo.         .xx         ..o         ...
====        ====        ====        ====        ====
..x         x.x         .o.         x.x         xox
.xo         .x.         .oo         .oo         .o.
xo.         oo.         .xx         ..o         .o.
====        ====        ====        ====        ====
            x.x         .o.         x.x
            .xo         .oo         xoo
            oox         xxx         ..o
            ====        ====        ====
                                    xox
                                    xoo
                                    ..o
                                    ====
                                    xox
                                    xoo
                                    x.o
                                    ====
```

The first two games (from the left) are played with the policy going first. The policy wins both. The last three are played with the policy going second. The policy wins twice and loses once. The policy still fails to recognize that the opponent is about to win, and fails to block the opponent.

Problem 2

Since the opponent is improving at the same time as the agent, one cannot really infer from the learning curve (figure3).

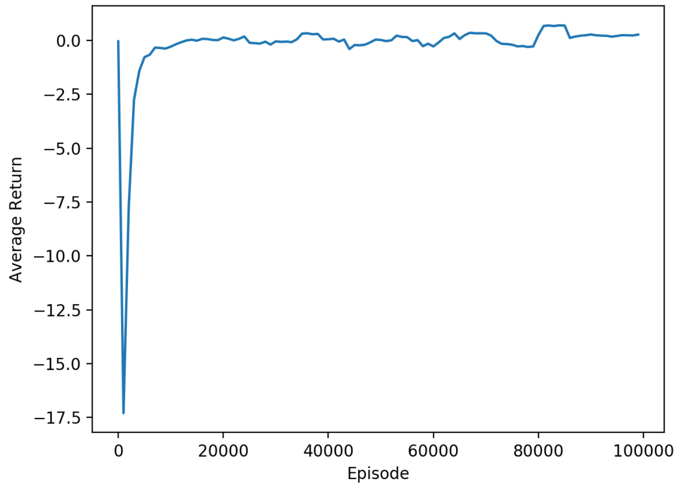

Figure 3: Self-Play Learning Curve

The agent performs fairly well against a random player when it is the first player, with a win rate around 80 percent.However, when it is the second player, the win rate drops to around 40 percent. This makes sense since in a game of tic-tac-toe, the player that goes first have a significant advantage. The win rate versus episode graph is shown in the figure below (figure4)

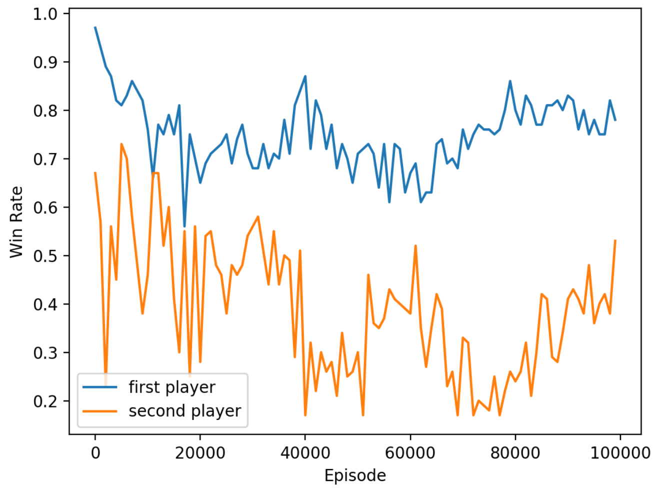

Figure 4: Win Rate vs. Episode (play against random)

However, if the agent plays against itself, it tends to tie instead of win. This is reasonable since both the agent and the opponent were trained in the same way and they would both try to win with the same method; therefore, most of the game would just tie. A snippet of the result output is shown below:

```
# Games: 100    Wins: 2         Ties: 98        Losses: 0       First Player: X
# Games: 100    Wins: 2         Ties: 98        Losses: 0       First Player: X
# Games: 100    Wins: 1         Ties: 99        Losses: 0       First Player: X
# Games: 100    Wins: 4         Ties: 96        Losses: 0       First Player: X
# Games: 100    Wins: 0         Ties: 100       Losses: 0       First Player: X
# Games: 100    Wins: 0         Ties: 99        Losses: 1       First Player: O
# Games: 100    Wins: 0         Ties: 95        Losses: 5       First Player: O
# Games: 100    Wins: 1         Ties: 96        Losses: 3       First Player: O
# Games: 100    Wins: 1         Ties: 99        Losses: 0       First Player: O
# Games: 100    Wins: 2         Ties: 95        Losses: 3       First Player: O
```

The return for winning the game was increased as an attempt to improve the win rate. However, the output figure (figure 5) says otherwise. One can see that the two lines are mirrored which was not expected. The reason for this is that since both players both really want to win now and they were both trained the same, they know each other's weaknesses. Therefore, when the opponent gets to play first, it would completely destroy the agent and vice versa.

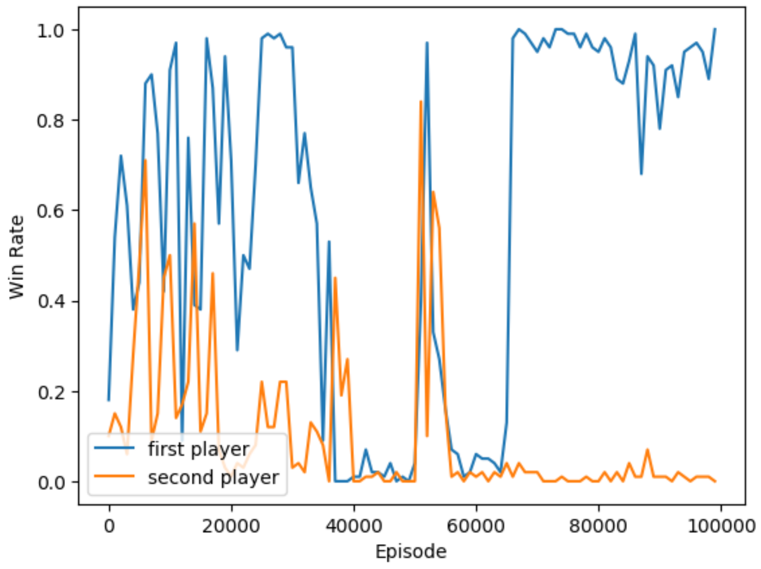

Figure 5: Win Rate vs. Episode (play against itself with new return values)

Sample game output when the agent plays against a random player:

```
Game 80
...
.x.
o..
====
x..
.x.
oo.
====
x..
.x.
oox
====
Game 60
...
ox.
...
====
..x
ox.
.o.
====
..x
ox.
xo.
====
Game 40
...
.o.
...
====
..x
.o.
o..
====
x.x
.o.
o.o
====
xxx
.o.
o.o
====
Game 20
.o.
.x.
...
====
.ox
.x.
o..
====
xox
ox.
o..
====
xox
ox.
o.x
====
Game 0
...
.x.
.o.
====
..x
.xo
.o.
====
..x
.xo
xo.
====
```

Sample game output when the agent plays against itself:

```
Game 80
..o
.x.
...
====
x.o
.x.
..o
====
x.o
ox.
x.o
====
x.o
oxx
xoo
====
xxo
oxx
xoo
====
Game 60
...
.o.
...
====
..x
.o.
o..
====
xox
.o.
o..
====
xox
.ox
oo.
Game 50
...
.o.
...
====
..x
.o.
o..
====
xox
.o.
```

```
o..
====
xox
.ox
o.o
====
xox
.ox
o.o
====
xox
xox
ooo
====
Game 40
...
.o.
...
====
x.o
.o.
...
====
x.o
oo.
x..
====
xxo
oo.
xo.
====
xxo
ooo
xox
====
Game 20
...
.o.
...
====
x.o
.o.
...
====
x.o
.o.
o.x
====
```

One can see that the games are now longer when it is playing against itself.