Одбрана пројекта

Основи Машинског учења

Филип Милић

Септембар 2024

Тема

Имплементација карташке игре таблић са четири играча

Уопштено о окружењу

- Окружење имплементирано у класи Tablic
- Пружа функционалности попут:
- Провере да ли је потез валидан, налажења свих валидних потеза
- Одигравање потеза, ажурирање игре, представљање стања и акција...

Напомене

- У оргиналној реализацији таблића са 2 играча не прави се разлика између знакова
- Дакле "мала"двојка и десетка ромб вреде 0 и 1 поен респективно
- Како би се задржала конзистенција са претходним радом исто је урађено и у верзији са 4 играча

Карте

- Карте се у окружењу представљају и нумерички и векторски
- Вектор од 13 елемената може представљати талон, узете карте, руку..

Стање

- Вектор акције у стању је вектор nx1, n=106.
- Вектор акције у стању садржи векторе:
- Талона, карте играча на потезу, однете карте сва 4 играча
- Одигране карте, однетих карата
- Такође садржи и
- Индикатор играча који је последњи носио, индикатор тренутног играча

Алгоритам учења

- Коришћен је алгоритам дубоког "Q"учења
- Оргинална верзија користи Double DQN са бафером за понављање искустава
- У верзији са четири играча коришћен исти због компарабилности

Поставка за тренирање

- Користи се e-похлепна политика како би се мотивисало истраживање агента
- Хиперпараметри задржани константим

```
# Number of episodes
EPISODES = 50000
SAVE_FREQ = 5000

# Update frequency
UPDATE_FREQ = 5
# Switch nets frequency
SWITCH_FREQ = 50
# Plot frequency
PLOT_FREQ=1000

# Epsilon
EPSILON_START = 1.0
EPSILON_END = 0.1
EPSILON_DECAY = (EPSILON_START - EPSILON_END) / EPISODES
```

Поставка за тренирање

- Агент игра сам против себе
- Награде су представљене са бројем поена који је донео неки потез
- Тестирани хиперпараметри:
- alpha, gamma и величина mini-batch-a

Архитекуре

- Испробане су различите архитектуре дубоке Q мреже
- Са резидуалним конекцијама, са суперпонираним шумом..
- Основна:

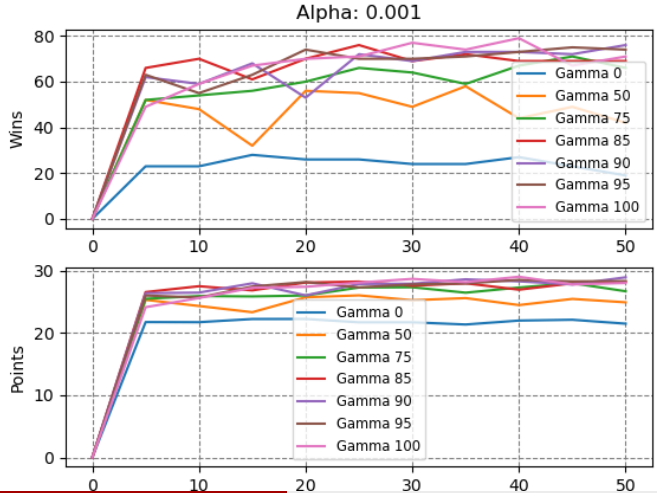
```
inputSize = 80
outputSize = 1
self.reluStack = nn.Sequential(
    nn.Linear(inputSize, inputSize*2),
    nn.ReLU(),
    nn.Linear(inputSize*2, inputSize*2),
    nn.ReLU(),
    nn.Linear(inputSize*2, outputSize))
```

Евалуација

- Евалуација на идентичном скупу шпилова карата
- По 100 одиграних партија
- Против похлепног играча
- Са заменом положаја играча
- Више поена у збиру оба положаја = Побендик

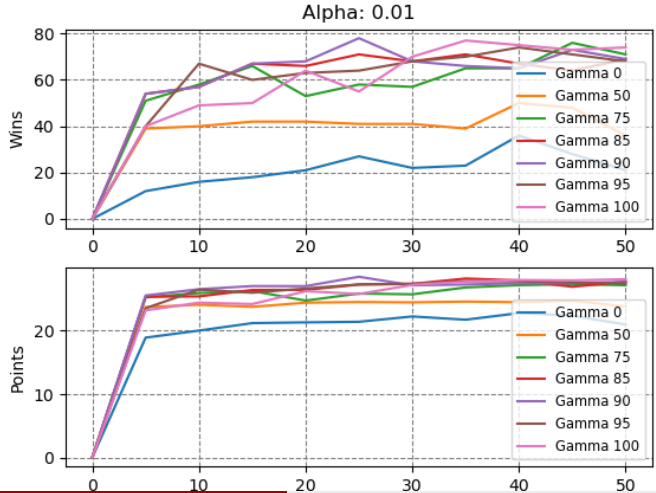
Исходи појединих тренинг сесија

- **Поставка хиперпараметара као у оргиналној верзији:**



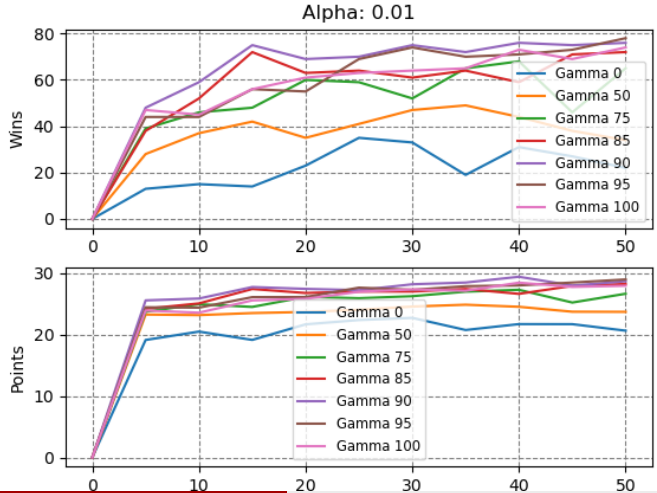
Исходи појединих тренинг сесија

- **alpha=0.01:**



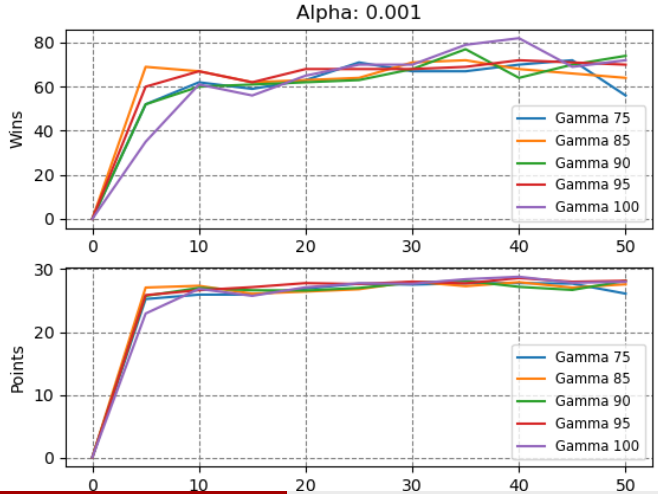
Исходи појединих тренинг сесија

- **mini-batch size=2048, alpha=0.01:**



Исходи појединих тренинг сесија

- **Оргинална поставка хиперпараметара, тимски вектор однетих у стању**



Резултати

- **Статистика обученог агента у игри са 2 играча**

Player	wins	draws	losses	player points	opponent points
Greedy Player	0	1000	0	26.214	26.214
Reinforced Player $\gamma = 0.00$	535	32	433	26.828	25.341
Reinforced Player $\gamma = 0.50$	592	31	377	27.912	25.359
Reinforced Player $\gamma = 0.75$	620	33	347	28.617	24.703
Reinforced Player $\gamma = 0.85$	685	39	276	29.257	24.158
Reinforced Player $\gamma = 0.90$	665	32	303	29.141	24.124
Reinforced Player $\gamma = 0.95$	748	32	220	29.894	23.347
Reinforced Player $\gamma = 1.00$	732	38	230	30.170	23.232

Резултати

- **Статистика обученог агента у игри са 4 играча**

Gamma	wins	draws	losses	player points	opponent points
0.0	24	3	73	22.04	26.36
0.5	51	3	46	25.22	23.89
0.75	61	9	30	26.48	23.61
0.85	74	0	26	27.5	22.69
0.9	73	1	26	28.47	22.67
0.95	64	7	29	27.49	23.55
1.0	68	3	29	27.28	23.68

Занимљивост

- **Похлепни играч који не избацује насумично карту на талон када нема ношења, већ задржава штихове, надиграва играча који то чини**

Крај

- **Хвала на пажњи.**