

OpenAIThon

일시 : 17-09-17

장소 : 서울시립대

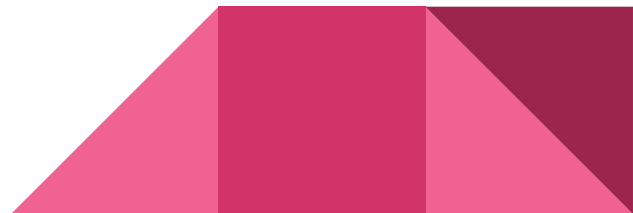
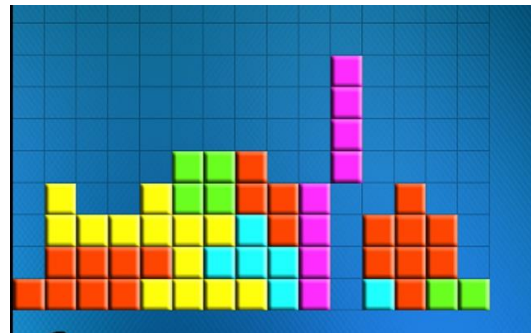
강촌수련원

팀장 : 최순호

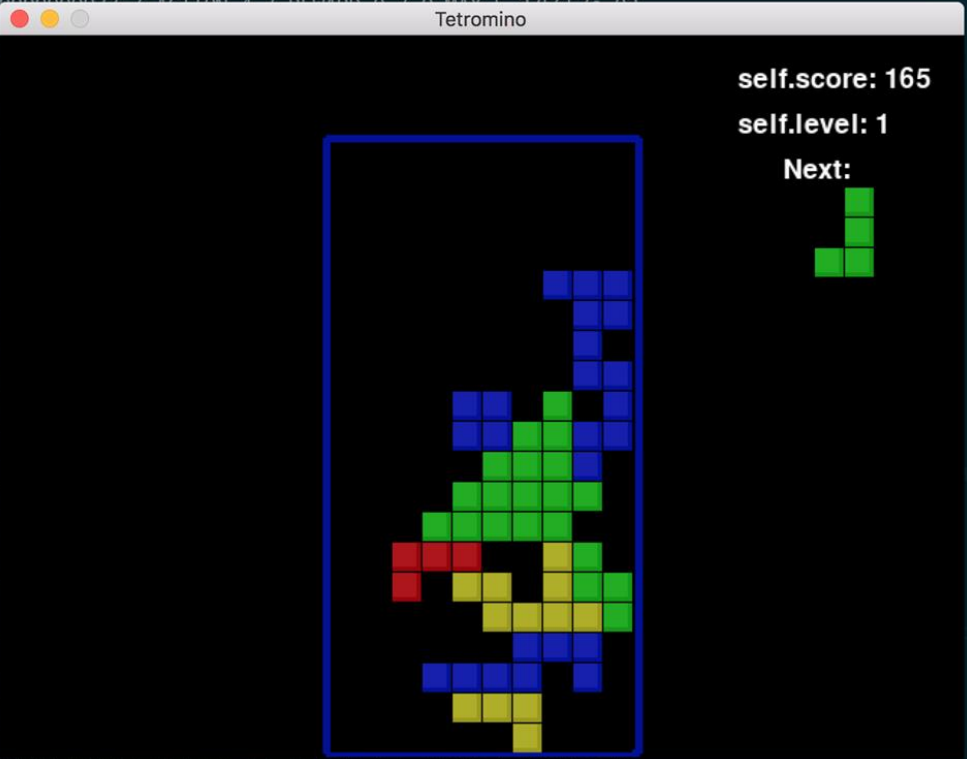
팀원

- 김장현
- 임동건
- 송남주
- 류강현

Motivation

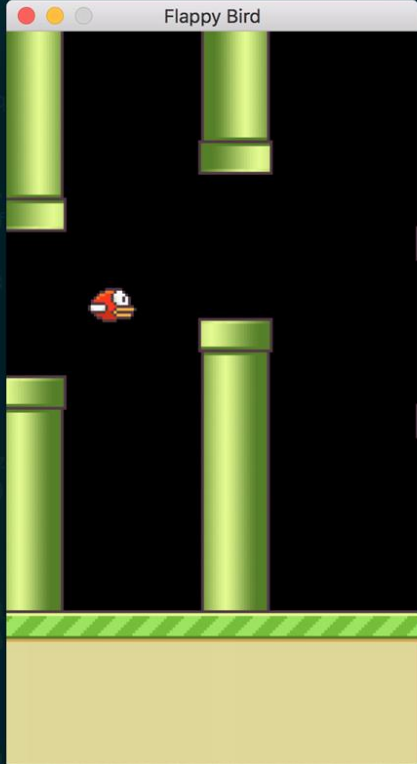


```
1. python3 deep_q_network.py (Python)
TIMESTEP 677 / STATE explore / EPSILON 0.6655999999999977 / ACTION 2 / REWARD 0 / Q_MAX 4.955547e-02
TIMESTEP 678 / STATE explore / EPSILON 0.6636999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 679 / STATE explore / EPSILON 0.6617999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 680 / STATE explore / EPSILON 0.6598999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 681 / STATE explore / EPSILON 0.6579999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 682 / STATE explore / EPSILON 0.6560999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 683 / STATE explore / EPSILON 0.6541999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 684 / STATE explore / EPSILON 0.6522999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 685 / STATE explore / EPSILON 0.6503999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 686 / STATE explore / EPSILON 0.6484999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 687 / STATE explore / EPSILON 0.6465999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 688 / STATE explore / EPSILON 0.6446999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 689 / STATE explore / EPSILON 0.6427999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 690 / STATE explore / EPSILON 0.6408999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 691 / STATE explore / EPSILON 0.6389999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 692 / STATE explore / EPSILON 0.6370999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 693 / STATE explore / EPSILON 0.6351999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 694 / STATE explore / EPSILON 0.6332999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 695 / STATE explore / EPSILON 0.6313999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 696 / STATE explore / EPSILON 0.6294999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 697 / STATE explore / EPSILON 0.6275999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 698 / STATE explore / EPSILON 0.6256999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 699 / STATE explore / EPSILON 0.6237999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 700 / STATE explore / EPSILON 0.6218999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 701 / STATE explore / EPSILON 0.6199999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 702 / STATE explore / EPSILON 0.6180999999999977 / ACTION 1 / REWARD 0 / Q_MAX 5.275657e-02
TIMESTEP 703 / STATE explore / EPSILON 0.6161999999999977 / ACTION 3 / REWARD 0 / Q_MAX 7.108092e-02
TIMESTEP 704 / STATE explore / EPSILON 0.6142999999999977 / ACTION 3 / REWARD 0 / Q_MAX 7.241760e-02
TIMESTEP 705 / STATE explore / EPSILON 0.6123999999999977 / ACTION 3 / REWARD 0 / Q_MAX 7.222950e-02
```



<https://github.com/asrivat1/DeepLearningVideoGames>

```
1. python3 deep_q_network.py (Python)
TIMESTEP 489 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.240166e+01
TIMESTEP 490 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.250980e+01
TIMESTEP 491 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.249850e+01
TIMESTEP 492 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.252997e+01
TIMESTEP 493 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.260642e+01
TIMESTEP 494 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.265901e+01
TIMESTEP 495 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.268556e+01
TIMESTEP 496 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.274777e+01
TIMESTEP 497 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.273537e+01
TIMESTEP 498 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.280637e+01
TIMESTEP 499 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.287524e+01
TIMESTEP 500 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.290447e+01
TIMESTEP 501 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.294386e+01
TIMESTEP 502 / STATE observe / EPSILON 0.0001 / ACTION 1 / REWARD 0.1 / Q_MAX 1.303273e+01
TIMESTEP 503 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.297798e+01
TIMESTEP 504 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 1 / Q_MAX 1.304882e+01
TIMESTEP 505 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.212112e+01
TIMESTEP 506 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.215228e+01
TIMESTEP 507 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.215507e+01
TIMESTEP 508 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.219193e+01
TIMESTEP 509 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.217243e+01
TIMESTEP 510 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.221924e+01
TIMESTEP 511 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.224975e+01
TIMESTEP 512 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.225022e+01
TIMESTEP 513 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.228287e+01
TIMESTEP 514 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.234788e+01
TIMESTEP 515 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.238458e+01
TIMESTEP 516 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.237753e+01
TIMESTEP 517 / STATE observe / EPSILON 0.0001 / ACTION 0 / REWARD 0.1 / Q_MAX 1.242579e+01
```



<https://github.com/yenchenlin/DeepLearningFlappyBird>

**우리도 이런
Game AI를
만들어 보자!**

Used Open Source



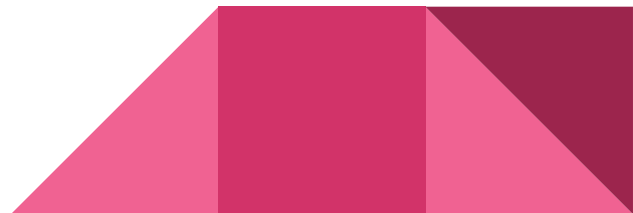
Python



TensorFlow



Git

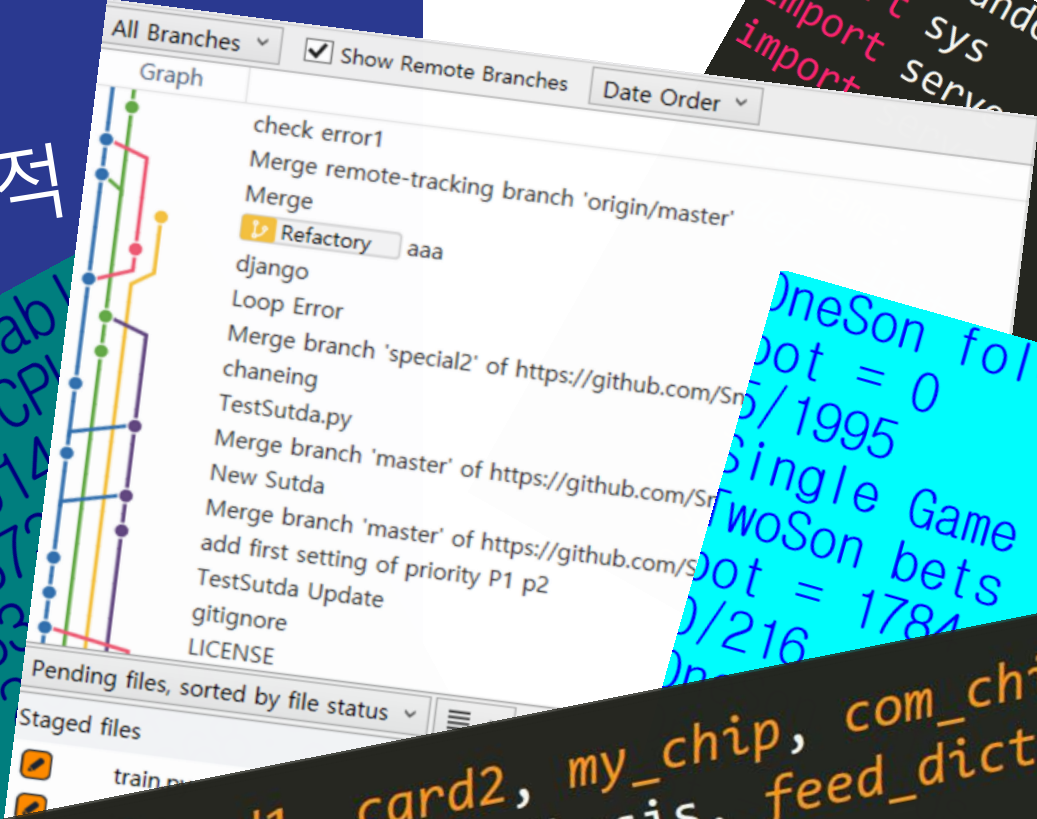


Model Info

- 나의 카드 1, 나의 카드 2, 남은 자신의 칩, 남은 상대방의 칩, 나의 베팅 칩 수, 상대의 베팅 칩 수, 베팅 횟수
- 이번 베팅에서 몇 개의 칩을 베팅할 것인지 결정
- batch_size=1000, layer=4. shape=128, step=5001
- 초기값 설정 xavier_initializer
- 활성화 함수 ReLU
- AdamOptimizer

그냥 좋다는 건 다 썼다.

AI와 나, 성장의 흔적



def betting(card1, card2, my_chip, com_chip, return sess.run(hypothesis, feed_dict={X:

Single Game Done Two Son Victories Random

Step 1.

랜덤 베팅을 하는
컴퓨터 대결로 만
든 데이터 셋


승리한 경우에 대
한 상태 값과 베팅
칩 수


dataset.csv


```
1 18,11,990,990,0,0,0,774
2 6,16,206,453,0,1321,3,206
3 8,3,388,312,0,1280,9,388
4 12,14,1678,302,0,0,10,1323
5 12,6,12,1268,0,700,7,12
6 13,1,990,990,0,0,0,897
7 13,1,93,4,897,986,2,93
8 15,14,1992,0,0,0,3,1727
9 7,10,831,540,159,450,2,830
10 8,7,662,978,0,340,3,531
11 10,17,111,454,0,1415,7,111
12 8,1,1775,118,27,60,17,521
13 14,11,990,990,0,0,0,812
14 6,20,168,890,0,922,3,168
15 18,11,1100,880,0,0,4,1039
16 5,1,990,990,0,0,0,964
17 16,12,6,460,0,1518,4,6
18 2,6,1980,0,0,0,16,1197
19 6,19,990,990,0,0,0,647
20 6,19,343,97,647,893,2,343
21 8,5,204,313,0,1463,8,204
22 2,6,990,990,0,0,0,919
```


Step 2.


서로 다른 두 개
의 데이터 셋으로
학습시킨 두가지
모델 생성


 dataset_new.csv


 dataset.csv


 dataset2_new.csv


 dataset2.csv


 hw3!.c


 LICENSE


 model.data-00000-of-00001


 model.index


 model.meta


 model2.data-00000-of-00001


 model2.index


 model2.meta

 README.md

 serve.py

 serve2.py

 train.py

 train2.py

Step 3.

두 모델끼리 다시
게임을 진행하여
새로운 데이터 셋
을 만듦

dataset_new.csv

```
1 6,14,654,654,336,336,2,336
2 20,19,517,849,255,359,4,359
3 20,17,540,820,268,352,4,352
4 17,6,650,650,340,340,2,340
5 1,7,654,654,336,336,2,336
6 9,15,1682,298,0,0,0,483
7 19,9,650,650,340,340,2,340
8 12,5,1700,280,0,0,0,487
9 9,1,651,651,339,339,2,339
10 8,13,653,653,337,337,2,337
11 11,18,1694,286,0,0,0,486
12 19,9,650,650,340,340,2,340
13 7,12,1680,300,0,0,5,474
14 20,9,650,650,340,340,2,340
15 2,5,1690,290,0,0,0,475
16 11,6,651,651,339,339,2,339
17 20,18,651,651,339,339,2,339
18 15,20,475,1047,0,458,13,458
19 4,1,652,652,338,338,2,338
20 1,6,654,654,336,336,2,336
21 6,14,654,654,336,336,2,336
22 2,20,656,656,334,334,2,334
23 20,16,1678,302,0,0,0,495
24 12,5,651,651,339,339,2,339
25 15,3,1698,282,0,0,0,490
26 12,7,651,651,339,339,2,339
27 18,5,1678,302,0,0,5,486
28 7,6,652,652,338,338,2,338
```

dataset2_new.csv

```
1 17,14,1682,298,0,0,0,492
2 9,20,621,706,318,335,4,335
3 1,17,630,704,316,330,11,330
4 4,9,1674,306,0,0,0,477
5 19,20,1672,308,0,0,0,494
6 4,12,1704,276,0,0,0,478
7 9,16,1684,296,0,0,0,483
8 3,16,1819,161,0,0,0,479
9 8,18,1692,288,0,0,0,482
10 18,17,830,170,359,621,5,359
11 10,6,1674,306,0,0,0,484
12 16,10,1670,310,0,0,0,491
13 12,2,1674,306,0,0,0,486
14 15,12,1678,302,0,0,0,490
15 4,15,1682,298,0,0,0,478
16 1,14,831,175,355,619,5,355
17 9,10,1674,306,0,0,0,483
18 7,2,528,840,257,355,9,355
19 4,5,1676,301,0,0,0,477
20 2,13,654,318,336,672,3,336
21 8,1,1682,298,0,0,0,482
22 12,11,830,171,358,621,5,358
23 14,11,652,314,338,676,3,338
24 14,5,1690,290,0,0,0,489
25 5,2,1676,304,0,0,0,478
26 2,7,1584,396,0,0,0,473
27 10,6,653,316,337,674,3,337
28 6,19,1706,274,0,0,0,480
```

