

05/23 機器學習 打磚塊 正式比賽

張君豪

New Original Rule (ver.2)

- 公式算出球的落點
- 再讓板子移到接的到球的地方

ml_play_second_new_original_rule.py

```
1 """The template of the main script of the machine learning process
2 """
3
4 from .. import communication as comm
5 from ..communication import SceneInfo, GameInstruction
6 import pickle, time
7 import numpy as np
8 #import os.path
9
10 def ml_loop():
11     """The main loop of the machine learning process
12     This loop is run in a separate process, and communicates with the game process.
13     Note that the game process won't wait for the ml process to generate the
14     GameInstruction. It is possible that the frame of the GameInstruction
15     is behind of the current frame in the game process. Try to decrease the fps
16     to avoid this situation.
17     """
18
19     # === Here is the execution order of the loop === #
20     # 1. Put the initialization code here.
21     a=0
22     b=0
23     c=93
24     d=93
25     #c=100
26     #d=100
27
28     # 2. Inform the game process that ml process is ready before start the loop.
29     frame_ind=[]
30     ballpos=[]
31     platform=[]
32     instruct=[]
33     des_test=[]
34     LL=[]
35     LL2=[]
36     filename="neigh_knn_model_0416_knn_game123_fps24.sav"
37     load_model=pickle.load(open(filename, 'rb'))
```

```

38  → #filename="neigh_knn_model_0416_knn_game123_fps24.sav" CRLF
39  → #filepath=os.path.join(os.path.dirname(__file__), filename) CRLF
40  → #load_model=pickle.load(open(filepath, 'rb')) CRLF
41  → comm.ml_ready() CRLF
42  CRLF
43  → #3. Start an endless loop. CRLF
44  → while True: CRLF
45  →     → #3.1. Receive the scene information sent from the game process. CRLF
46  →     → scene_info=comm.get_scene_info() CRLF
47  →     → frame_ind.append(scene_info.frame) CRLF
48  →     → ballpos.append(scene_info.ball) CRLF
49  →     → platform.append(scene_info.platform) CRLF
50  →     → #all_infor=[frame_ind, ballpos, platform, instruct] CRLF
51  →     → all_infor=[frame_ind, ballpos, platform, instruct, LL, LL2] CRLF
52  →     → #inp_temp=np.array([scene_info.ball[0], scene_info.ball[1], scene_info.platform[0]]) CRLF
53  →     → #input=inp_temp[np.newaxis, :] CRLF
54  →     → #3.2. If the game is over or passed, the game process will reset CRLF
55  →     → #.....the scene immediately and send the scene information again. CRLF
56  →     → #.....Therefore, receive the reset scene information. CRLF
57  →     → #.....You can do proper actions, when the game is over or passed. CRLF
58  →     → if scene_info.status == SceneInfo.STATUS_GAME_OVER or \ CRLF
59  →     →     → scene_info.status == SceneInfo.STATUS_GAME_PASS: CRLF
60  →     →     → #save game data CRLF
61  →     →     → fname='game_infor'+time.strftime("%m_%d_%H_%M_%S", time.localtime())+'.pickle' CRLF
62  →     →     → file=open(fname, 'wb') CRLF
63  →     →     → pickle.dump(all_infor, file) CRLF
64  →     →     → file.close() CRLF
65  →     →     → #fname2='des_infor'+time.strftime("%m_%d_%H_%M_%S", time.localtime())+'.pickle' CRLF
66  →     →     → #file2=open(fname2, 'wb') CRLF
67  →     →     → #pickle.dump(des_test, file2) CRLF
68  →     →     → #file2.close() CRLF
69  →     → scene_info=comm.get_scene_info() CRLF
70  CRLF
71  →     → #3.3. Put the code here to handle the scene information CRLF
72  →     → a=c CRLF
73  →     → b=d CRLF
74  →     → c=scene_info.ball[0] CRLF

```

```
71  →→→→→# 3.3. Put the code here to handle the scene informationCRLF
72  →→→→→a=cCRLF
73  →→→→→b=dCRLF
74  →→→→→c=scene_info.ball[0]CRLF
75  →→→→→d=scene_info.ball[1]CRLF
76  →→→→→m=((d-b)/(c-a))CRLF
77  →→→→→L=(400-d+c*m)/mCRLF
78  →→→→→LL.append(L)CRLF
79  →→→→→px=0CRLF
80  →→→→→###inp_temp=np.array([scene_info.ball[0], scene_info.ball[1], scene_info.platform[0], K, ((d-b)/(c-a)), (c-a), (d-b), K2, a, b])CRLF
81  →→→→→#inp_temp=np.array([L, scene_info.platform[0]])CRLF
82  →→→→→#input=inp_temp[np.newaxis, :.]CRLF
83  CRLF
84  →→→→→# 3.4. Send the instruction for this frame to the game process.....CRLF
85  →→→→→if L>=0 and L<=200:CRLF
86  →→→→→→→→→→→L=LCRLF
87  →→→→→elif L<0:CRLF
88  →→→→→→→→→→→#L=(-L)%200CRLF
89  →→→→→→→→→→→L=-LCRLF
90  →→→→→→→→→→→num=L//200CRLF
91  →→→→→→→→→→→rr=L%200CRLF
92  →→→→→→→→→→→if num%2==1:CRLF
93  →→→→→→→→→→→→→→→→→L=200-rrCRLF
94  →→→→→→→→→→→else:CRLF
95  →→→→→→→→→→→→→→→→→L=rrCRLF
96  →→→→→→→→→→→elif L>200:CRLF
97  →→→→→→→→→→→→→→→→→#L=200-(L-200)%200CRLF
98  →→→→→→→→→→→→→→→→→L=L-200CRLF
99  →→→→→→→→→→→→→→→→→num=L//200CRLF
100 →→→→→→→→→→→→→→→→→rr=L%200CRLF
101 →→→→→→→→→→→→→→→→→if num%2==1:CRLF
102 →→→→→→→→→→→→→→→→→→→→→→→→L=rrCRLF
103 →→→→→→→→→→→→→→→→→else:CRLF
104 →→→→→→→→→→→→→→→→→→→→→→→→L=200-rrCRLF
105 →→→→→→→→→→→→→→→→→CRLF
106 →→→→→→→→→→→→→→→→→LL2.append(L)CRLF
107 →→→→→→→→→→→→→→→→→CRLF
```

ml_play_second_new_original_rule.py

```
110      >>> elif L<0 and L>-200: CR LF
111      >>> # L := -L CR LF
112      >>> elif L>200 and L<400: CR LF
113      >>> # L := 400-L CR LF
114      >>> CR LF
115      >>> if L>=0 and L<20: CR LF
116      >>>     >px=0 CR LF
117      >>> elif L>=20 and L<40: CR LF
118      >>>     >px=10 CR LF
119      >>> elif L>=40 and L<60: CR LF
120      >>>     >px=30 CR LF
121      >>> elif L>=60 and L<80: CR LF
122      >>>     >px=50 CR LF
123      >>> elif L>=80 and L<100: CR LF
124      >>>     >px=70 CR LF
125      >>> elif L>=100 and L<120: CR LF
126      >>>     >px=90 CR LF
127      >>> elif L>=120 and L<140: CR LF
128      >>>     >px=110 CR LF
129      >>> elif L>=140 and L<160: CR LF
130      >>>     >px=130 CR LF
131      >>> elif L>=160 and L<180: CR LF
132      >>>     >px=150 CR LF
133      >>> elif L>=180 and L<200: CR LF
134      >>>     >px=160 CR LF
135      >>> CR LF
136      >>> CR LF
137      >>> if scene_info.platform[0]<px: CR LF
138      >>>     >comm.send_instruction(scene_info.frame, GameInstruction.CMD_RIGHT) CR LF
139      >>>     >instruct.append(1) CR LF
140      >>> elif scene_info.platform[0]>px: CR LF
141      >>>     >comm.send_instruction(scene_info.frame, GameInstruction.CMD_LEFT) CR LF
142      >>>     >instruct.append(-1) CR LF
143      >>> else: CR LF
144      >>>     >comm.send_instruction(scene_info.frame, GameInstruction.CMD_NONE) CR LF
145      >>>     >instruct.append(0) CR LF
146      >>> CR LF
```

Python file

length: 4,547 lines: 157

Ln: 1 Col: 1 Sel: 0 | 0

Windows (CR LF) UTF-8

INS

用pickle取data放進.csv檔

- Feature X:
 - 球的落點、板子的x座標
- Predict Y:
 - 移動板子的指令 (0、1、-1)

```
pickle1.py - MLGame-master_5th x pickle2.py - MLGame-master_5th x pickle1.py - Desktop\...\5月23研究 new rule (ver2) and new train x p < > ⚙
```

```
1# -*- coding: utf-8 -*-
2"""
3Created on Thu Mar 14 16:01:47 2019
4
5@author: VraainsHacker
6"""
7
8import pickle
9with open("2019-05-23_01-45-53_game7.pickle", "rb") as f:
10    data_list = pickle.load(f)
11
12
13# save each information seperately
14Frame=[]
15Status=[]
16Ballposition=[]
17PlatformPosition=[]
18Bricks=[]
19
20for i in range(0,len(data_list)):
21    Frame.append(data_list[i].frame)
22    Status.append(data_list[i].status)
23    Ballposition.append(data_list[i].ball)
24    PlatformPosition.append(data_list[i].platform)
25    Bricks.append(data_list[i].bricks)
26
27import numpy as np
28
29PlatX=np.array(PlatformPosition)[:,:0][:, np.newaxis]
30PlatX_next=PlatX[1:,:]
31instruct=(PlatX_next-PlatX[0:len(PlatX_next),0][:,np.newaxis])/5
```

Permissions: **RW** End-of-lines: **C**


```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Thu Mar 14 16:01:47 2019
4
5 @author: VrainHacker
6 """
7
8 import pickle
9 with open("game_infor05_23_01_45_53_game7.pickle", "rb") as f:
10     data_list = pickle.load(f)
11
12
13 # save each information seperately
14 datax1=data_list[1]
15 datax2=data_list[2]
16 datax3=data_list[3]
17 data_LL=data_list[4]
18 data_LL2=data_list[5]
19
20
21 #import numpy as np
22 #x1 = np.array([10, 20, 30], float)
23 #print("shape of x1 is ", x1.shape)
24 #print(x1)
25 #
26 #x2 = x1[:, np.newaxis]
27 #print("shape of x2 is ", x2.shape)
28 #print(x2)
29 #
30 #x3 = x1[np.newaxis, :]
31 #print("shape of x3 is ", x3.shape)
32 #print(x3)

```



微軟正黑體 10

L1					
	A	B	C	D	E
1	index	X(200)	plate_x	instruction	
2		0	75	-1	game3 start
3		0	70	-1	
4		0	65	-1	
5		0	60	-1	
6		0	55	-1	
7		0	50	-1	
8		0	45	-1	
9		0	40	-1	
10		0	35	-1	
11		0	30	-1	
12		0	25	-1	
13		0	20	-1	
14		0	15	-1	
15		0	10	-1	
16		89.5714285714286	5	1	
17		7	10	-1	
18		7	5	-1	
19		7	0	0	
20		7	0	0	
21		7	0	0	
22		7	0	0	
23		7	0	0	
24		7	0	0	
25		7	0	0	
26		7	0	0	
27		7	0	0	

Load data and train model

- 用knn
- Feature X:
 - 球的落點、板子的x座標
- Predict Y:
 - 移動板子的指令 (0、1、-1)

```
8 import pandas as pd #for handling .csv files
9 import numpy as np
10
11 customer_data =pd.read_csv("game_12367_all_data.csv")
12 customer_data.head(10) # show first ten samples of data
13
14 x=customer_data.iloc[:,1:3].values
15 y=customer_data.iloc[:,3:4].values
16
17
18 from sklearn.model_selection import train_test_split
19 x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
20
21 ### train your model here
22 #from xxx import ooo_model
23 from sklearn.neighbors import KNeighborsClassifier
24 #ooo=ooo_model()
25 neigh = KNeighborsClassifier(n_neighbors=1)
26 #ooo.fit(x_train,y_train)
27 #neigh.fit(x_train,y_train)
28 neigh.fit(x,y)
29 #ooo.predict(x_test)
30 y_knn=neigh.predict(x)
31 # check the acc to see how well you've trained the model
32 #acc=?
33 from sklearn.metrics import accuracy_score
34 acc=accuracy_score(y_knn,y)
35
36 ### save model
37 import pickle
38
39 #filename="ooo_example0401.sav"
40 filename="neigh_knn_game12367_0523.sav"
41 pickle.dump(neigh, open(filename, 'wb'))
```

ml_play with machine learning model

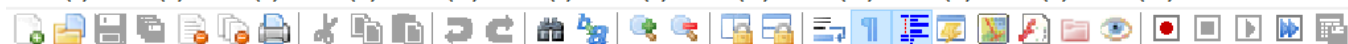


ml_play_pass_game12367_knn_new.py

```

1  """The template of the main script of the machine learning processCRLF
2  """CRLF
3  CRLF
4  from .. import communication as commCRLF
5  from ..communication import SceneInfo, GameInstructionCRLF
6  import pickle, timeCRLF
7  import numpy as npCRLF
8  #import os.pathCRLF
9  def ml_loop():CRLF
10     """The main loop of the machine learning processCRLF
11     CRLF
12     -> This loop is run in a separate process, and communicates with the game process.CRLF
13     CRLF
14     -> Note that the game process won't wait for the ml process to generate theCRLF
15     -> GameInstruction. It is possible that the frame of the GameInstructionCRLF
16     -> is behind of the current frame in the game process. Try to decrease the fpsCRLF
17     -> to avoid this situation.CRLF
18     """CRLF
19     CRLF
20     -> # .=== Here is the execution order of the loop .=== #CRLF
21     -> # 1. Put the initialization code here.CRLF
22     -> a=0CRLF
23     -> b=0CRLF
24     -> c=93CRLF
25     -> d=93CRLF
26     -> #c=100CRLF
27     -> #d=100CRLF
28     -> # 2. Inform the game process that ml process is ready before start the loop.CRLF
29     -> frame_ind=[]CRLF
30     -> ballpos=[]CRLF
31     -> platform=[]CRLF
32     -> instruct=[]CRLF
33     -> des_test=[]CRLF
34     -> #LL=[]CRLF
35     -> #LL2=[]CRLF
36     -> filename="neigh_knn_game12367_0523.sav"CRLF
37     -> load_model=pickle.load(open(filename, 'rb'))CRLF

```



ml_play_pass_game12367_knn_new.py

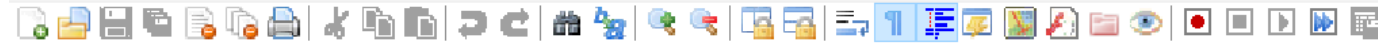
```

41  →comm.ml_ready() CR LF
42  CR LF
43  →# 3. Start an endless loop. CR LF
44  →while True: CR LF
45  →→→# 3.1. Receive the scene information sent from the game process. CR LF
46  →→→scene_info = comm.get_scene_info() CR LF
47  →→→frame_ind.append(scene_info.frame) CR LF
48  →→→ballpos.append(scene_info.ball) CR LF
49  →→→platform.append(scene_info.platform) CR LF
50  →→→#all_infor=[frame_ind, ballpos, platform, instruct] CR LF
51  →→→#all_infor=[frame_ind, ballpos, platform, instruct, LL, LL2] CR LF
52  →→→# inp_temp=np.array([scene_info.ball[0], scene_info.ball[1], scene_info.platform[0]]) CR LF
53  →→→# input=inp_temp[np.newaxis, :] CR LF
54  →→→# 3.2. If the game is over or passed, the game process will reset CR LF
55  →→→# ..... the scene immediately and send the scene information again. CR LF
56  →→→# ..... Therefore, receive the reset scene information. CR LF
57  →→→# ..... You can do proper actions, when the game is over or passed. CR LF
58  →→→if scene_info.status == SceneInfo.STATUS_GAME_OVER or \ CR LF
59  →→→→scene_info.status == SceneInfo.STATUS_GAME_PASS: CR LF
60  →→→→# save game data CR LF
61  →→→→# fname='game_infor'+time.strftime("%m_%d_%H_%M_%S", time.localtime())+'.pickle' CR LF
62  →→→→# file=open(fname, 'wb') CR LF
63  →→→→# pickle.dump(all_infor, file) CR LF
64  →→→→# file.close() CR LF
65  →→→→# fname2='des_infor'+time.strftime("%m_%d_%H_%M_%S", time.localtime())+'.pickle' CR LF
66  →→→→# file2=open(fname2, 'wb') CR LF
67  →→→→# pickle.dump(des_test, file2) CR LF
68  →→→→# file2.close() CR LF
69  →→→scene_info = comm.get_scene_info() CR LF
70  CR LF
71  →→→# 3.3. Put the code here to handle the scene information CR LF
72  →→→a=c CR LF
73  →→→b=d CR LF
74  →→→c=scene_info.ball[0] CR LF
75  →→→d=scene_info.ball[1] CR LF
76  →→→m=(d-b)/(c-a) CR LF
77  →→→L=(400-d+c*m)/m CR LF

```

C:\Users\VrainsHacker\Desktop\5月23研究 new rule (ver2) and new train\ml_play_pass_game12367_knn_new.py - Notepad++

檔案(F) 編輯(E) 搜尋(S) 檢視(V) 編碼(N) 語言(L) 設定(T) 工具(O) 巨集(M) 執行(R) 外掛(P) 視窗(W) ?



ml_play_pass_game12367_knn_new.py

```
85  if L>=0 and L<=200:CRLF
86      L = LCRLF
87  elif L<0:CRLF
88      #L = (-L)%200CRLF
89      L = -LCRLF
90      num = L//200CRLF
91      rr = L%200CRLF
92      if num%2==1:CRLF
93          L = 200-rrCRLF
94      else:CRLF
95          L = rrCRLF
96  elif L>200:CRLF
97      #L = 200-(L-200)%200CRLF
98      L = L-200CRLF
99      num = L//200CRLF
100     rr = L%200CRLF
101     if num%2==1:CRLF
102         L = rrCRLF
103     else:CRLF
104         L = 200-rrCRLF
105     CRLF
106     # 3.4. Send the instruction for this frame to the game process.....CRLF
107     CRLF
108     CRLF
109     inp_temp=np.array([L, scene_info.platform[0]])CRLF
110     input=inp_temp[np.newaxis, :]CRLF
111     CRLF
112     if load_model.predict(input)==1:CRLF
113         comm.send_instruction(scene_info.frame, GameInstruction.CMD_RIGHT)CRLF
114         instruct.append(1)CRLF
115     elif load_model.predict(input)==-1:CRLF
116         comm.send_instruction(scene_info.frame, GameInstruction.CMD_LEFT)CRLF
117         instruct.append(-1)CRLF
118     else:CRLF
119         comm.send_instruction(scene_info.frame, GameInstruction.CMD_NONE)CRLF
120         instruct.append(0)CRLF
121     CRLF
```

Python file

length: 4670 lines: 163

result

- Pass game 1 2 3 6 7