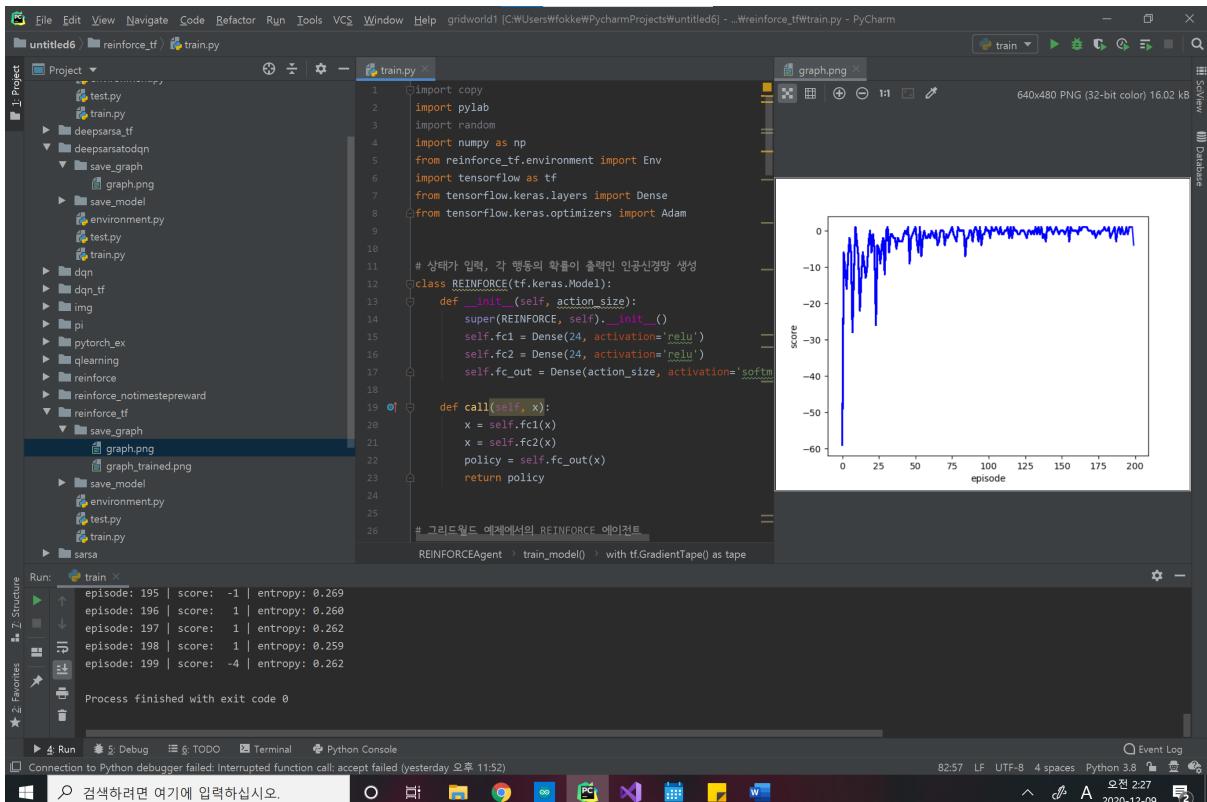


## 1. Grid world, Reinforce 예제의 tensorflow를 pytorch로 바꾸어보았습니다.

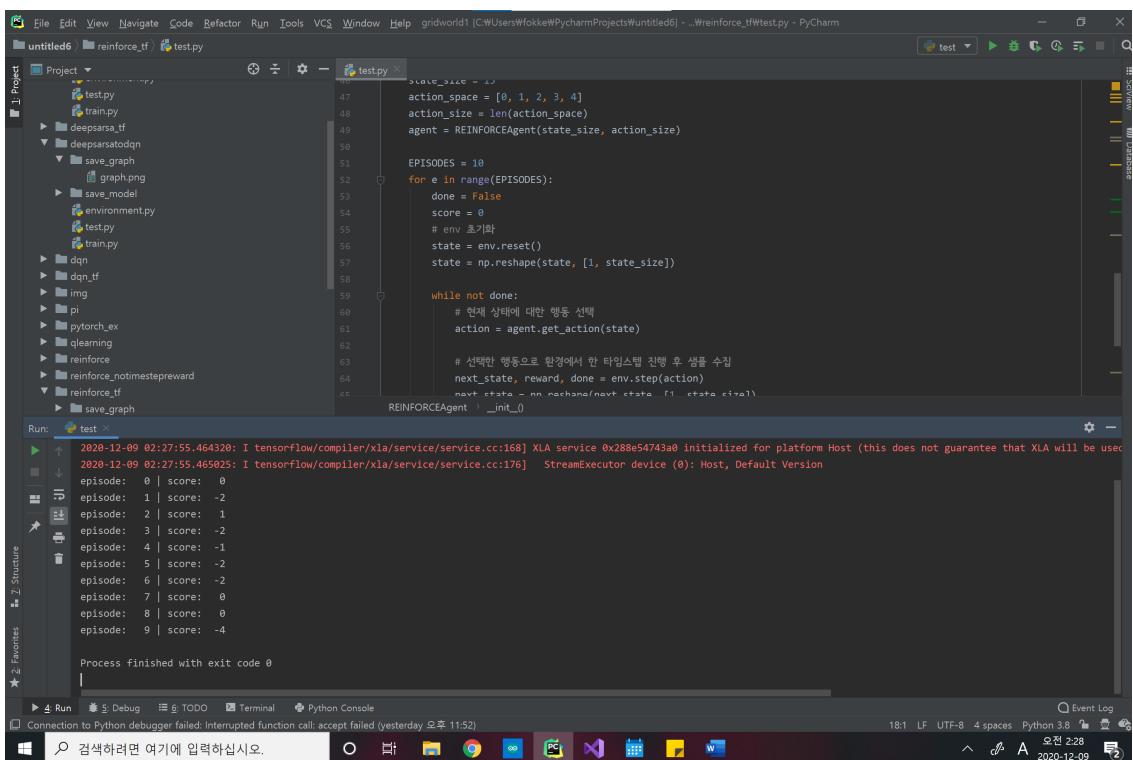


```
File Edit View Navigate Code Refactor Run Tools VCS Window Help gridworld1 [C:\Users\fokke\PycharmProjects\untitled6] - ...Reinforce_tf\train.py - PyCharm
untitled6 reinforcement_tf train.py
Project
  test.py
  train.py
  deepsarsa_tf
  save_graph
    graph.png
  save_model
    environment.py
    test.py
    train.py
  dqn
  dqn_tf
  img
  pi
  pytorch_ex
  qlearning
  reinforce
  reinforce_nostepreward
  reinforce_tf
    save_graph
      graph.png
      graph_trained.png
    save_model
      environment.py
      test.py
      train.py
  sarsa
Run: train
  episode: 195 | score: -1 | entropy: 0.269
  episode: 196 | score: 1 | entropy: 0.268
  episode: 197 | score: 1 | entropy: 0.262
  episode: 198 | score: 1 | entropy: 0.259
  episode: 199 | score: -4 | entropy: 0.262
Process finished with exit code 0
Favorites
Event Log
Connection to Python debugger failed: Interrupted function call: accept failed (yesterday 오후 11:52)
82:57 LF UTF-8 4 spaces Python 3.8 오전 2:27 2020-12-09
```

# 상태가 입력, 각 행동의 확률이 출력인 인공신경망 생성  
class REINFORCE(tf.keras.Model):  
 def \_\_init\_\_(self, action\_size):  
 super(REINFORCE, self).\_\_init\_\_()  
 self.fc1 = Dense(24, activation='relu')  
 self.fc2 = Dense(24, activation='relu')  
 self.fc\_out = Dense(action\_size, activation='softmax')  
  
 def call(self, x):  
 x = self.fc1(x)  
 x = self.fc2(x)  
 policy = self.fc\_out(x)  
 return policy

# 그리드월드 예제에서의 REINFORCE 에이전트  
REINFORCEAgent > train\_model() > with tf.GradientTape() as tape

tensorflow로 구현한 train 결과입니다.



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help gridworld1 [C:\Users\fokke\PycharmProjects\untitled6] - ...Reinforce_tf\test.py - PyCharm
untitled6 reinforcement_tf test.py
Project
  test.py
  train.py
  deepsarsa_tf
  save_graph
    graph.png
  save_model
    environment.py
    test.py
    train.py
  dqn
  dqn_tf
  img
  pi
  pytorch_ex
  qlearning
  reinforce
  reinforce_nostepreward
  reinforce_tf
    save_graph
  Run: test
  2020-12-09 02:27:55.464320: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x288e54743a0 initialized for platform Host (this does not guarantee that XLA will be used
  2020-12-09 02:27:55.465925: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
  episode: 0 | score: 0
  episode: 1 | score: -2
  episode: 2 | score: 1
  episode: 3 | score: -2
  episode: 4 | score: -1
  episode: 5 | score: -2
  episode: 6 | score: -2
  episode: 7 | score: 0
  episode: 8 | score: 0
  episode: 9 | score: -4
Process finished with exit code 0
Favorites
Event Log
Connection to Python debugger failed: Interrupted function call: accept failed (yesterday 오후 11:52)
18:1 LF UTF-8 4 spaces Python 3.8 오전 2:28 2020-12-09
```

```
state_size = 1
action_space = [0, 1, 2, 3, 4]
action_size = len(action_space)
agent = REINFORCEAgent(state_size, action_size)

EPISODES = 10
for e in range(EPISODES):
    done = False
    score = 0
    # env 초기화
    state = env.reset()
    state = np.reshape(state, [1, state_size])

    while not done:
        # 현재 상태에 대한 행동 선택
        action = agent.get_action(state)

        # 선택한 행동으로 환경에서 한 타임스텝 진행 후 샘플 수집
        next_state, reward, done = env.step(action)
        next_state = np.reshape(next_state, [1, state_size])
```

Tensorflow로 학습한 model을 사용한 test 결과입니다.

The screenshot shows the PyCharm IDE interface. The project structure on the left includes files like 'gridworld1', 'reinforce', and 'reinforce\_nointimestepreward'. The current file is 'train.py' at line 133. The code implements a REINFORCE algorithm with entropy regularization. A line plot titled 'graph.png' is displayed on the right, showing the score over 200 episodes. The score fluctuates between -25 and 0.

```

    score += reward
    agent.append_sample(state, action, reward)

    state = next_state

if done:
    # 에피소드마다 정책신경망 업데이트
    entropy = agent.train_model()
    # 에피소드마다 학습 결과 출력
    print("episode: {:3d} | score: {:3d} | entropy: {:.3f}".format(e, score, entropy))

    scores.append(score)
    episodes.append(e)
    pylab.plot(episodes, scores, 'b')
    pylab.xlabel("episode")
    pylab.ylabel("score")
    pylab.savefig("./save_graph/graph.png")

```

이는 pytorch로 바꿔준 train결과입니다

The screenshot shows the PyCharm IDE interface. The project structure on the left includes files like 'gridworld1', 'reinforce', and 'reinforce\_nointimestepreward'. The current file is 'test.py' at line 37. The code defines a REINFORCEAgent class that loads a pre-trained model from a file. The terminal output shows the scores for 9 episodes, which are significantly higher and more stable than those in the training run, ranging from -5 to 0.

```

        nn.Softmax(),
    )

    def forward(self, x):
        policy = self.layer(x)
        return policy

# 그리드월드 예제에서의 REINFORCE 에이전트
class REINFORCEAgent:
    def __init__(self, state_size, action_size):
        # 상태의 크기와 행동의 크기 정의
        self.state_size = state_size
        self.action_size = action_size

        self.model = REINFORCE(self.action_size)
        self.model.load_state_dict(torch.load('./save_model/' + 'model_state_dict.pt'))

    # 정책신경망으로 행동 선택
    def get_action(self, state):
        REINFORCEAgent

```

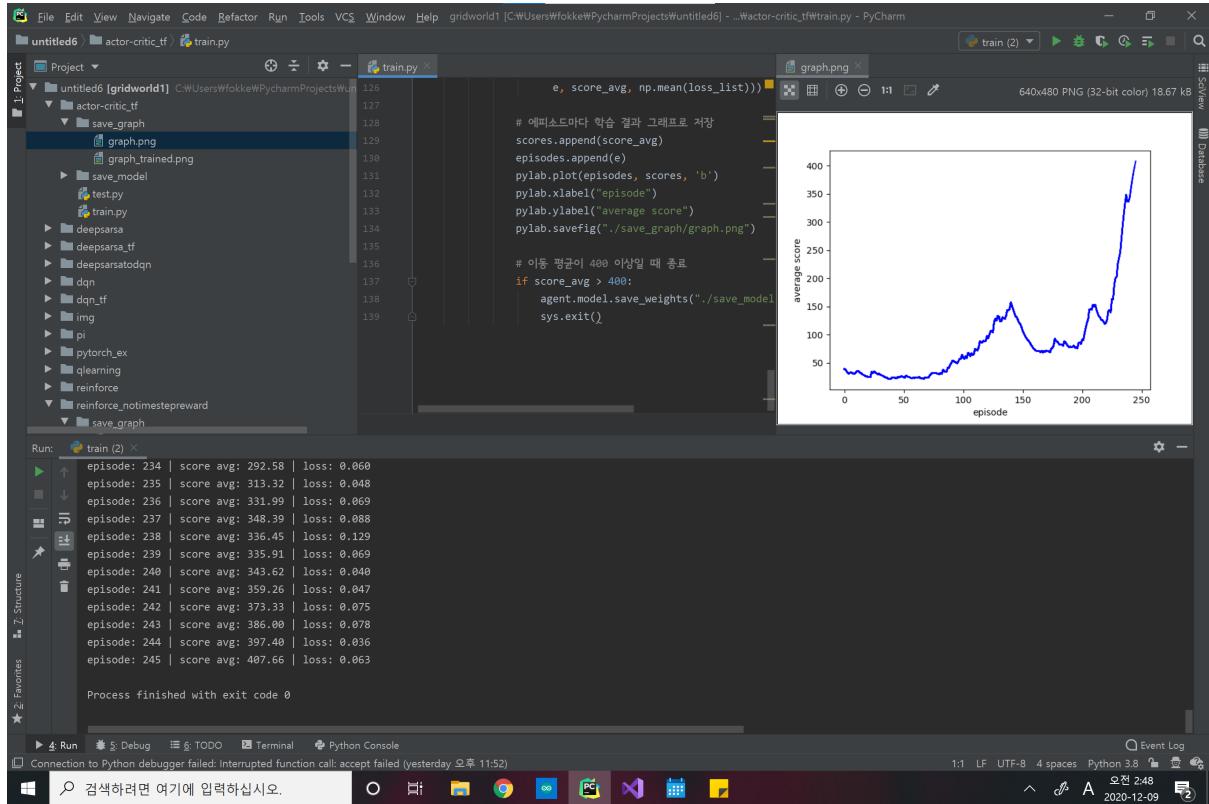
Process finished with exit code 0

이는 위에서 학습한 model을 사용한 test결과입니다

Tensorflow와 pytorch의 train결과와 test결과를 보면 pytorch로 바꿔준 결과가 score가 늦게 올라

값을 확인 하였습니다. Pytorch가 학습까지 필요한 episode가 좀더 많아보였습니다.

## Cartpole example



위는 cartpole 예제를 tensorflow로 구현한 train 결과입니다.

```

32 # 기트풀 예제에서의 액터-크리틱(A2C) 에이전트
33
34 class A2CAgent:
35     def __init__(self, action_size):
36         # 행동의 크기 정의
37         self.action_size = action_size
38
39         # 정책신경망과 가치신경망 생성
40         self.model = A2C(self.action_size)
41         self.model.load_weights("./save_model/model")
42
43         # 정책신경망의 출력을 받아 확률적으로 행동을 선택
44         def get_action(state):
45             policy, _ = self.model(state)
46             policy = np.array(policy[0])
47             return np.random.choice(self.action_size, 1, p=policy)[0]
48
49     if __name__ == "__main__":
50         A2CAgent()

```

To Change all layers to have dtype float64 by default, call `tf.keras.backend.set\_floatx('float64')`. To change just this layer, pass dtype='float64' to the layer constructor. If you are using TensorFlow 2.x, you can use tf.dtypes.float64 instead.

episode: 0 | score: 500  
episode: 1 | score: 500  
episode: 2 | score: 500  
episode: 3 | score: 500  
episode: 4 | score: 500  
episode: 5 | score: 500  
episode: 6 | score: 500  
episode: 7 | score: 500  
episode: 8 | score: 500  
episode: 9 | score: 500

Process finished with exit code 0

이는 위에서 학습한 model로 test한 결과입니다. 모두 500점을 넘어 잘 학습되었음을 확인하였습니다.

이제 pytorch로 바꾸어준 model을 보겠습니다.

하지만 이를 파이토치로 바꾸어 보았을 때는 어디에선가 문제가 발생해 점수가 안 오르는 현상을 관찰하였다. 밑에 gridworld를 pytorch,actorcritic으로 구현했을 때 학습이 잘되는 것으로 보아, get action, train model, main 함수에는 문제가 없지만 model의 구조만이 달랐기 때문에 model의 구조에 문제가 있는 것 같았다. 하지만 model을 이런저런 방향(Dropout, batchnormalization, lr 바꾸기, model의 깊이 바꿔보기, clip norm 값 수정 등등)으로 바꾸어 보았지만 결국 점수는 올라가지 않았다.

```

scores, episodes = [], []
EPISODES = 200
for e in range(EPISODES):
    done = False
    score = 0
    # env 초기화
    state = env.reset()
    state = torch.FloatTensor(np.reshape(state, [1, state_size]))
    while not done:
        # 현재 상태에 대한 행동 선택
        action = agent.get_action(state)

        # 선택한 행동으로 환경에서 한 타임스텝 진행 후 샘플 수집
        next_state, reward, done = env.step(action)
        next_state = torch.FloatTensor(np.reshape(next_state, [1, state_size]))
        score += reward
        agent.append_sample(state, action, reward)

        state = next_state
    if done:
        # 에피소드마다 정책신경망 업데이트
        entropy = agent.train_model()
        # 에피소드마다 학습 결과 출력
        print("episode: {} | score: {} | entropy: {}".format(e, score, entropy))
        scores.append(score)
        episodes.append(e)

```

Run: train (2) × train (1) ×

episode	score	entropy
195	0	0.301
196	0	0.272
197	-1	0.291
198	-6	0.298
199	1	0.288

Process finished with exit code 0

이는 time step마다 -0.1의 reward를 주지 않은 train 결과이다

```

scores, episodes = [], []
EPISODES = 200
for e in range(EPISODES):
    done = False
    score = 0
    # env 초기화
    state = env.reset()
    state = torch.FloatTensor(np.reshape(state, [1, state_size]))
    while not done:
        # 현재 상태에 대한 행동 선택
        action = agent.get_action(state)

        # 선택한 행동으로 환경에서 한 타임스텝 진행 후 샘플 수집
        next_state, reward, done = env.step(action)
        next_state = torch.FloatTensor(np.reshape(next_state, [1, state_size]))
        score += reward
        reward -= 0.1
        agent.append_sample(state, action, reward)

        state = next_state
    if done:
        # 에피소드마다 정책신경망 업데이트
        entropy = agent.train_model()
        # 에피소드마다 학습 결과 출력
        print("episode: {} | score: {} | entropy: {}".format(e, score, entropy))
        scores.append(score)
        episodes.append(e)

```

Run: train (2) × train (3) ×

episode	score	entropy
195	0	0.249
196	0	0.242
197	1	0.245
198	0	0.252
199	1	0.224

Process finished with exit code 0

위는 pytorch로 바꾼 이후 time step마다 -0.1의 reward를 준 model의 train 결과이다.

그래프를 비교해보면 아래의 모델이 훨씬 학습이 잘되며 정확도가 높은 것을 육안으로 확인할 수 있다.

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help gridworld1 [C:\Users\fokke\PycharmProjects\untitled6] - ...reinforce_notimestepreward\test.py - PyCharm
untitled6 reinforcement_notimestepreward test.py
Project train.py test.py
def forward(self, x):
    policy = self.layer(x)
    return policy

# 그리드월드 예제에서의 REINFORCE 에이전트
class REINFORCEAgent:
    def __init__(self, state_size, action_size):
        # 상태의 크기와 행동의 크기 정의
        self.state_size = state_size
        self.action_size = action_size

        self.model = REINFORCE(self.action_size)
        self.model.load_state_dict(torch.load('./save_model/' + 'model_state_dict.pt'))

    # 정책신경망으로 행동 선택
    def get_action(self, state):
        policy = self.model(state)[0]

        with torch.no_grad():
            policy = policy.numpy()
        return np.random.choice(self.action_size, 1, p=policy)[0]

    if __name__ == "__main__":
        # 환경과 에이전트 생성
        env = Env(render_speed=0.05)

REINFORCEAgent
Run: train (2) test (2)
episode: 4 | score: -4
episode: 5 | score: -4
episode: 6 | score: 0
episode: 7 | score: -2
episode: 8 | score: -3
episode: 9 | score: -1
Process finished with exit code 0
Event Log
Connection to Python debugger failed: Interrupted function call: accept failed (yesterday 오후 11:52)
오전 2:18 2020-12-09

```

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help gridworld1 [C:\Users\fokke\PycharmProjects\untitled6] - ...reinforce\test.py - PyCharm
untitled6 reinforcement test.py
Project graph.png
def forward(self, x):
    policy = self.layer(x)
    return policy

# 그리드월드 예제에서의 REINFORCE 에이전트
class REINFORCEAgent:
    def __init__(self, state_size, action_size):
        # 상태의 크기와 행동의 크기 정의
        self.state_size = state_size
        self.action_size = action_size

        self.model = REINFORCE(self.action_size)
        self.model.load_state_dict(torch.load('./save_model/' + 'model_state_dict.pt'))

    # 정책신경망으로 행동 선택
    def get_action(self, state):
        policy = self.model(state)[0]

        with torch.no_grad():
            policy = policy.numpy()
        return np.random.choice(self.action_size, 1, p=policy)[0]

    if __name__ == "__main__":
        # 환경과 에이전트 생성
        env = Env(render_speed=0.05)
        state_size = 15
        action_space = [0, 1, 2, 3, 4]
        action_size = len(action_space)

REINFORCEAgent
Run: test (1)
input = module(input)
episode: 0 | score: -1
episode: 1 | score: -1
episode: 2 | score: 1
episode: 3 | score: 1
episode: 4 | score: -7
episode: 5 | score: 1
episode: 6 | score: 1
episode: 7 | score: -1
Process finished with exit code 0
Event Log
Connection to Python debugger failed: Interrupted function call: accept failed (yesterday 오후 11:52)
오전 2:19 2020-12-09

```

test결과도 아래의 모델을 보면 1의 score로 높지만 위의 모델은 아직 부족한 것을 확인하였다.

### 3 . grid world를 actor critic을 이용한 model로 바꾸어준 train 결과이다

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** untitled6 [gridworld1]
- Code Editor:** train.py (highlighted)
- Run:** train (3) - Shows the output of the script:

```
episode: 193 | score: 1
episode: 194 | score: 1
episode: 195 | score: 1
episode: 196 | score: 1
episode: 197 | score: 1
episode: 198 | score: 1
episode: 199 | score: 1

Process finished with exit code 0
```

- Plot:** graph.png - A line graph showing the score over 200 episodes. The x-axis is labeled "episode" and ranges from 0 to 200. The y-axis is labeled "score" and ranges from -7 to 1. The plot shows a highly volatile line with frequent spikes between -6 and 1.
- Event Log:** Shows the command run: C:\Users\fokke\anaconda3\python.exe C:/Users/fokke/PycharmProjects/untitled6/ac\_gridworld/test.py

보면 매우 매우 좋은 성능으로 동작함을 확인하였다.

Cartpole과 달리 kernel initialize 기능을 빼주었다.

The screenshot shows the PyCharm IDE interface with the following details:

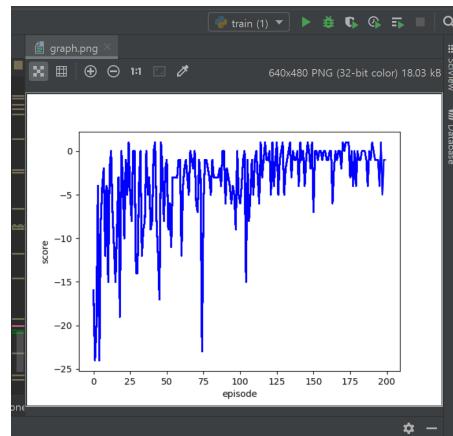
- Project:** untitled6 [gridworld1]
- Code Editor:** test.py (highlighted)
- Run:** test - Shows the output of the script:

```
episode: 0 | score: 1
episode: 1 | score: 1
episode: 2 | score: 1
episode: 3 | score: 1
episode: 4 | score: 1
episode: 5 | score: 1
episode: 6 | score: 1
episode: 7 | score: 1
episode: 8 | score: 1
episode: 9 | score: 1

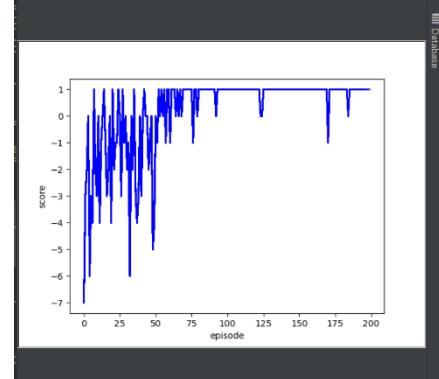
Process finished with exit code 0
```

- Event Log:** Shows the command run: C:\Users\fokke\anaconda3\python.exe C:/Users/fokke/PycharmProjects/untitled6/ac\_gridworld/test.py

test결과도 매우 훌륭하게 나왔다.



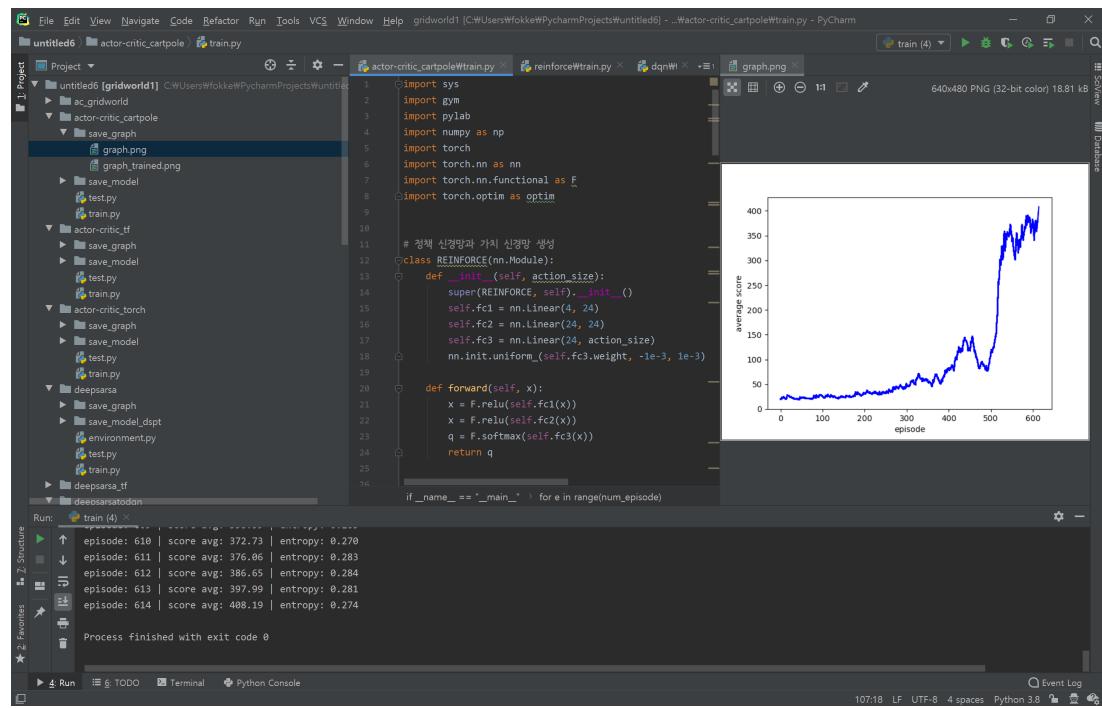
그래프 1



그래프2

그래프 1은 grid world를 reinforce로 학습시켰을 때이며 actor critic으로 학습시킨 그래프2와 비교해 보면 actor critic의 점수 그래프가 훨씬 안정적이게 1값에 수렴함을 확인하였다.

#### 4. cart pole 예제를 reinforce를 활용한 알고리즘으로 바꾸어보았다.

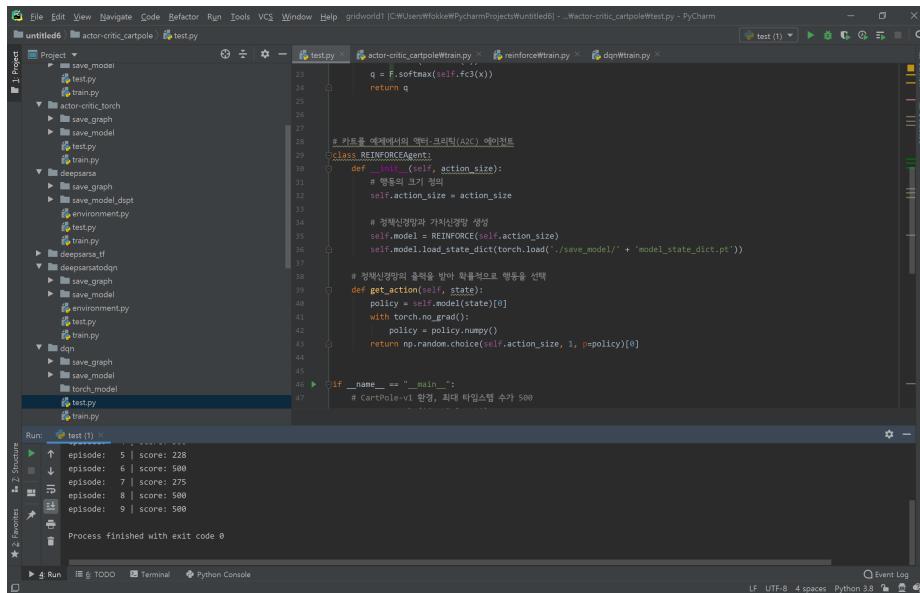


train결과를 보면 614 episode만에 avg score 400을 넘어 성공적으로 학습되었다.

위에 cartpole을 Actor critic, pytorch로 구현 했을때는 점수가 오르지 않았었는데 reinforce, pytorch로 구현한 결과 성공적으로 구현이 되었음을 확인하였다. 결국 Actor critic의 model에 문제가 있음을 간접적으로 추론할수 있었다. 이전에 gridworld에서 쓴 train\_model함수를 그

대로 사용하였으며 model만을 약간 바꾸어주었다.

교과서의 그래프와 비교해봤을 때 그래프의 형태가 굉장히 유사함을 확인할 수 있었다.



The screenshot shows the PyCharm IDE interface. The code editor displays a Python file named 'test.py' which contains code for a REINFORCE agent. The run output window shows the results of the 'test(1)' command, indicating success with a score of 500 over 9 episodes. The project structure on the left shows various files and folders related to the project, including 'actor-critic\_torch', 'deeparsa', 'dqn', and 'train.py'.

```
episode: 5 | score: 228
episode: 6 | score: 500
episode: 7 | score: 275
episode: 8 | score: 500
episode: 9 | score: 500
Process finished with exit code 0
```

위 model을 사용한 test결과도 성공적으로 500점이 나오는 것을 확인하였다.