

GSDS cluster 접속

- Slack 링크 :
<https://snugsds.slack.com/archives/C02EVU7MW93/p1660114170109369>
- VSCode에서 Remote로 cluster C 연결하기 :
<https://jstar0525.tistory.com/14>
- 위 슬랙링크에 attached된 클러스터 안내문에 있는 바와 같이 jupyter notebook 한번 실행해서 anaconda 설치되도록 하기

과제를 위한 가상환경설치 by terminal

- cluster내 본인 폴더에 assignment material 적절하게 위치시키고
- 해당 경로로 이동하여 아래 명령어로 가상환경만들어주기
(클러스터 이용하는 경우 "<your-system>" 자리에 "11.0" 입력하면 됨)
* 출처 : <https://web.stanford.edu/class/cs234/assignments/assignment2/index.html>

Create conda environment on your local system: replace <your-system> with your system, either mac or windows

```
cd starter_code_torch
conda env create -f cs234-torch-<your-system>.yaml
conda activate cs234-torch
```

- cmake가 없어서 atari-py 설치시 에러가 날 수 있음
(NameError: name 'cmd' is not defined)
 - 에러 난 이후에 가상환경 시작하고(conda activate cs234-torch)
 - pip install cmake 해주고
* 참고 링크 :
<https://velog.io/@ybkim95/NameError-name-cmd-is-not-defined>
 - pip install atari-py==0.2.6 으로 끝면 됨

Assignment 2-5 관련

- q5_linear_torch.py 파일 내에서
from torch.tensor import Tensor 는
from torch import Tensor
로 수정하면 에러 발생 안 함
- TBD add_optimizer 에서 learning rate는 어떻게 설정해야하는지?

Tensorboard 관련

- "login node"(중요!)의 terminal에서 아래 명령어 입력하여 사용 가능
tensorboard --logdir=results

Assignment 2-6 Nature QN implementation 관련

- architecture 참고 링크 (6쪽 Model Architecture):
<https://storage.googleapis.com/deepmind-data/assets/papers/DeepMindNature14236Paper.pdf>

The exact architecture, shown schematically in Fig. 1, is as follows. The input to the neural network consists of an $84 \times 84 \times 4$ image produced by the preprocessing map ϕ . The first hidden layer convolves 32 filters of 8×8 with stride 4 with the input image and applies a rectifier nonlinearity^{31,32}. The second hidden layer convolves 64 filters of 4×4 with stride 2, again followed by a rectifier nonlinearity. This is followed by a third convolutional layer that convolves 64 filters of 3×3 with stride 1 followed by a rectifier. The final hidden layer is fully-connected and consists of 512 rectifier units. The output layer is a fully-connected linear layer with a single output for each valid action. The number of valid actions varied between 4 and 18 on the games we considered.

- 논문에서 최초 input이 $84 \times 84 \times 4$ 인데, 과제에서는 $80 \times 80 \times 4$ 여서 hidden layer stride와 paddle을 어떻게 다뤄야하는지가 subtle함
- 아래 문구대로 padding을 설정하면 conv 통과한 후 output의 size 가 그대로 유지됨 (Conv2d layer의 argument를 padding='same'으로 하는 것과 동일할 듯)

```
class NatureQN(Linear):
    """
    Implementing DeepMind's Nature paper. Here are the relevant urls.
    https://storage.googleapis.com/deepmind-data/assets/papers/DeepMindNature14236Paper.pdf

    Model configuration can be found in the Methods section of the above paper.
    """

    def initialize_models(self):
        """Creates the 2 separate networks (Q network and Target network). The input
        to these models will be an img_height * img_width image
        with channels = n_channels * self.config.state_history

        1. Set self.q_network to be a model with num_actions as the output size
        2. Set self.target_network to be the same configuration self.q_network but initialized from scratch
        3. What is the input size of the model?

        To simplify, we specify the paddings as:
        .... (stride - 1) * img_height - stride + filter_size // 2

        Hints:
        1. Simply setting self.target_network = self.q_network is incorrect.
        2. The following functions might be useful
           - nn.Sequential
           - nn.Conv2d
           - nn.ReLU
           - nn.Flatten
           - nn.Linear
        """
        state_shape = list(self.env.observation_space.shape)
        img_height, img_width, n_channels = state_shape
        num_actions = self.env.action_space.n
```

아나콘다 가상환경 다루기

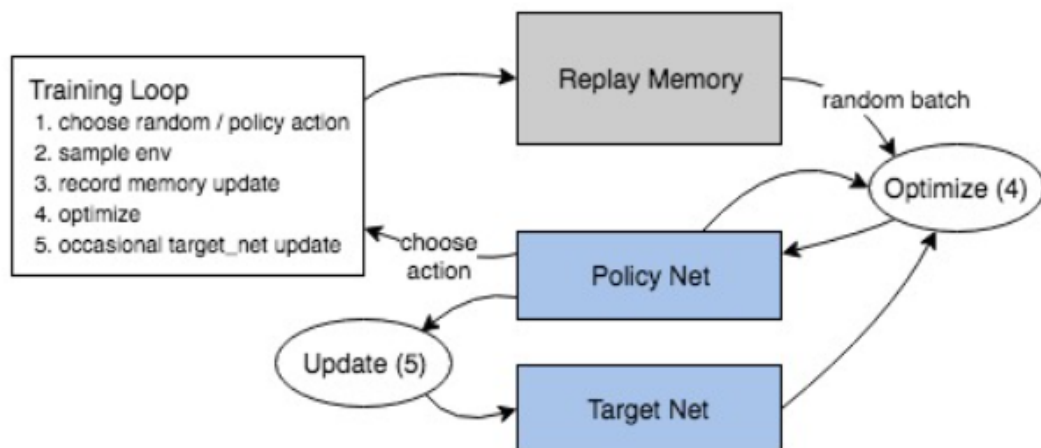
- <https://dev-woody.tistory.com/7>
-

한글 풀이 참고

- <https://cding.tistory.com/67>

참고 링크

- what is Gym Wrappers?
<https://alexandervandekleut.github.io/gym-wrappers/#:~:text=Wrapper%20that%20override%20how%20the.observation%20method%20of%20the%20environment.>
 - pytorch tutorial 내 DQN tutorial
-



https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html#sphx-glr-intermediate-reinforcement-q-learning-py