

Homework #1

Sept. 2024

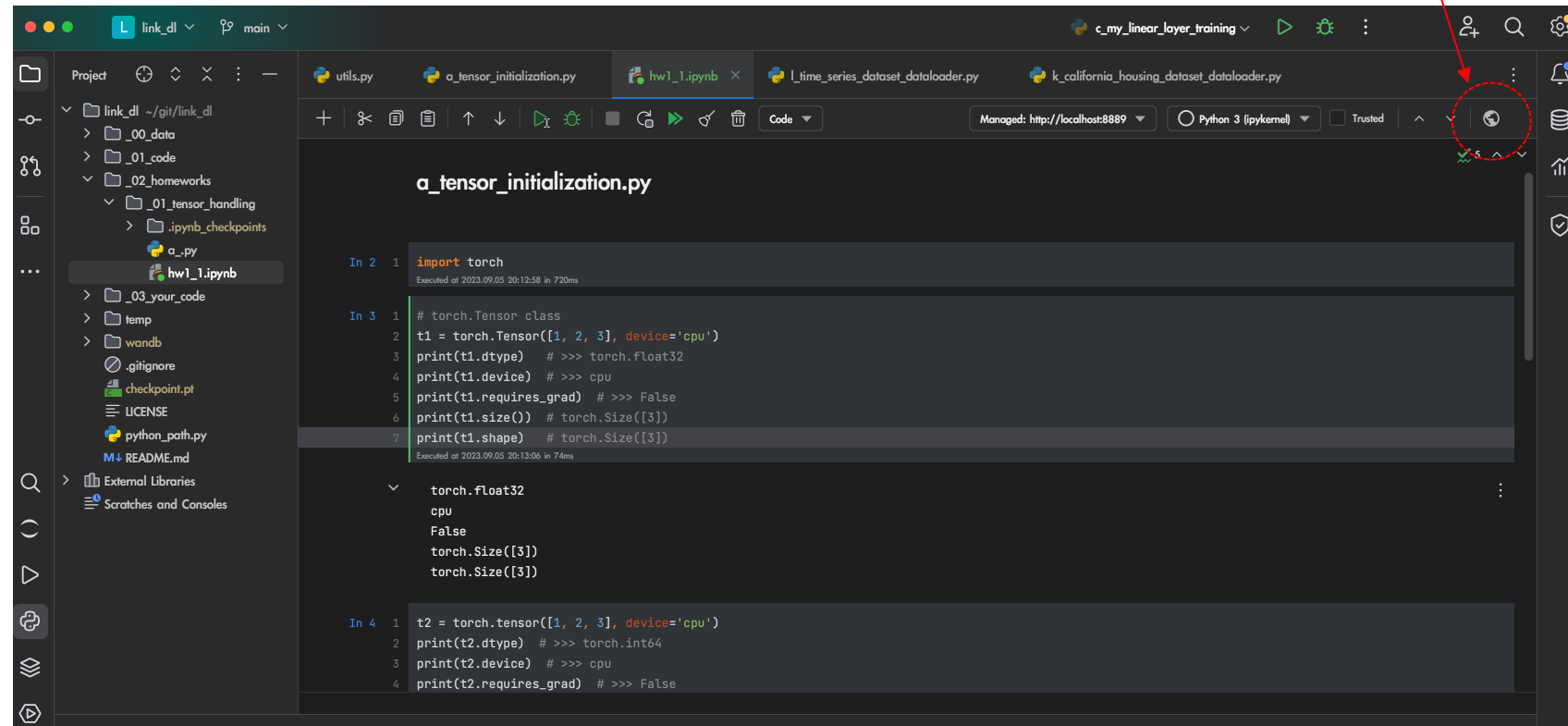
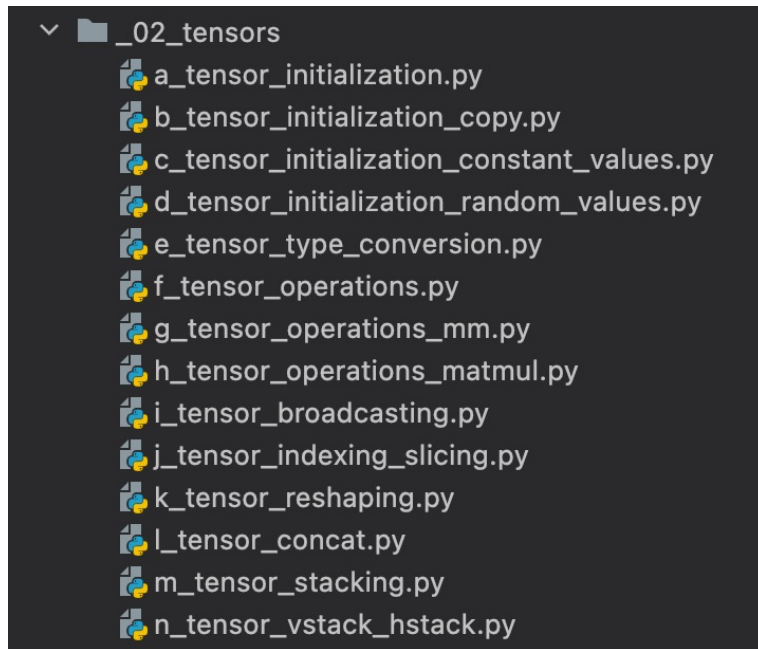
<http://link.koreatech.ac.kr>

1-1. Tensors (_02_Tensors) Codes - Practice

◆ _02_Tensors 폴더에서 제시하는 모든 예제를 ipynb에서 수행한 결과 제시

- Jupyter Notebook 설치
 - `pip install jupyter`
- Pycharm에서 Jupyter Notebook 실행가능하지만...
- 파일명: [hw1_1.ipynb](#)

Click!



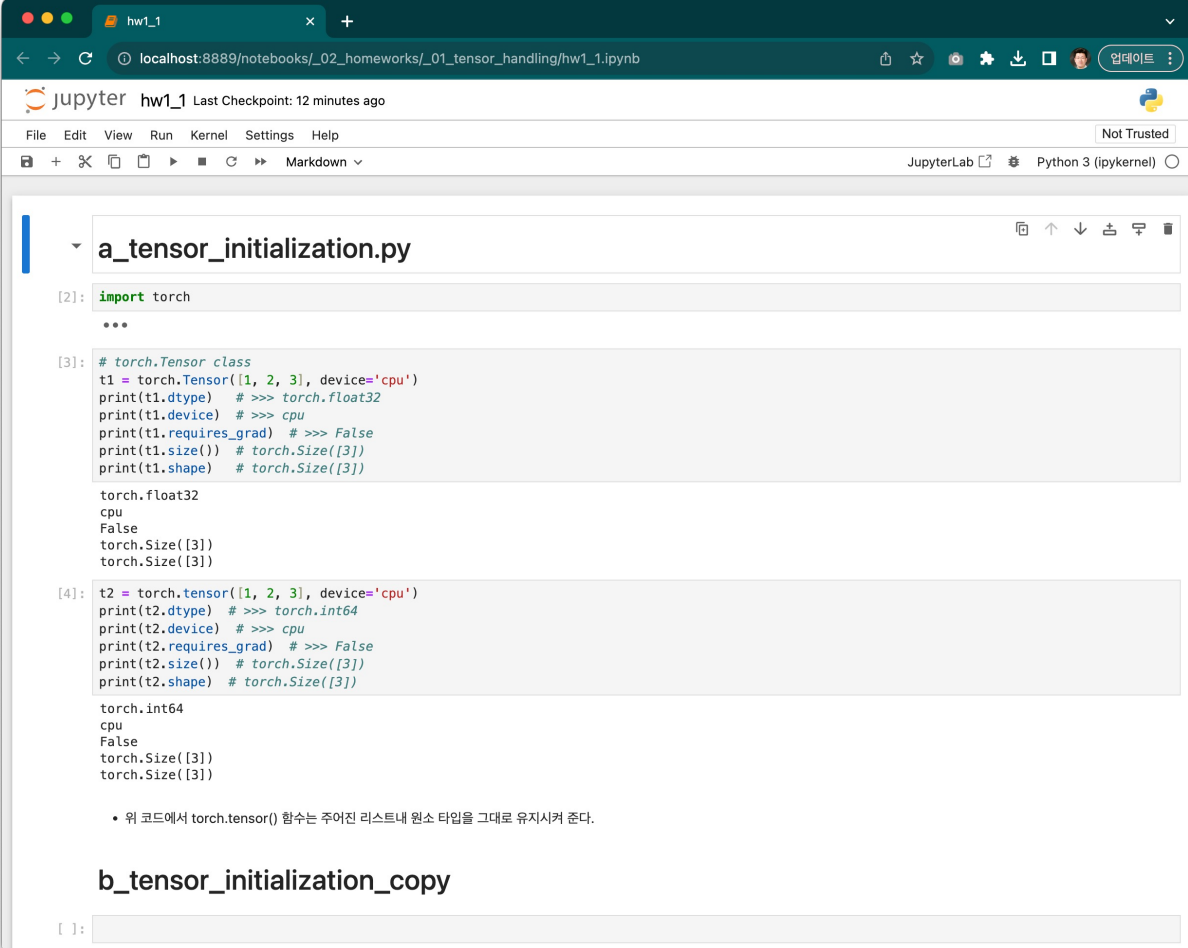
1. _02_Tensors - Self Practice

◆ _02_Tensors 폴더에서 제시하는 모든 예제를 ipynb에서 수행한 결과 제시

– Jupyter Notebook은 웹 브라우저에서 코딩 추천

– 단축키 적극 사용 하기

- ESC, a: 현재 셀 바로 위에 새로운 코드 셀 추가
- ESC, b: 현재 셀 바로 아래에 새로운 코드 셀 추가
- ESC, dd: 현재 셀 삭제
- ESC, m: 현재 셀을 markdown 셀로 변환
- ESC, y: 현재 셀을 code 셀로 변환
- ESC, c: 현재 셀 복사
- ESC, y: 현재 셀 잘라내기
- ESC, v: 복사하거나 잘라낸 셀을 붙여넣기



The screenshot shows a Jupyter Notebook window titled 'hw1_1' running on 'localhost:8889'. The notebook contains a file named 'a_tensor_initialization.py'. The code in the notebook is as follows:

```
[2]: import torch
...

[3]: # torch.Tensor class
t1 = torch.Tensor([1, 2, 3], device='cpu')
print(t1.dtype) # >>> torch.float32
print(t1.device) # >>> cpu
print(t1.requires_grad) # >>> False
print(t1.size()) # torch.Size([3])
print(t1.shape) # torch.Size([3])

torch.float32
cpu
False
torch.Size([3])
torch.Size([3])

[4]: t2 = torch.tensor([1, 2, 3], device='cpu')
print(t2.dtype) # >>> torch.int64
print(t2.device) # >>> cpu
print(t2.requires_grad) # >>> False
print(t2.size()) # torch.Size([3])
print(t2.shape) # torch.Size([3])

torch.int64
cpu
False
torch.Size([3])
torch.Size([3])
```

Below the code, there is a note: "• 위 코드에서 torch.tensor() 함수는 주어진 리스트내 원소 타입을 그대로 유지시켜 준다."

The notebook also shows a section titled 'b_tensor_initialization_copy' with an empty code cell below it.

1. _02_Tensors - Self Practice

◆ _02_Tensors 폴더에서 제시하는 모든 예제를 ipynb에서 수행한 결과 제시

– 1) 로컬에 저장된 hw1_1.ipynb 파일을 클라우드에 업로드 하기

- Github, Google drive 등...

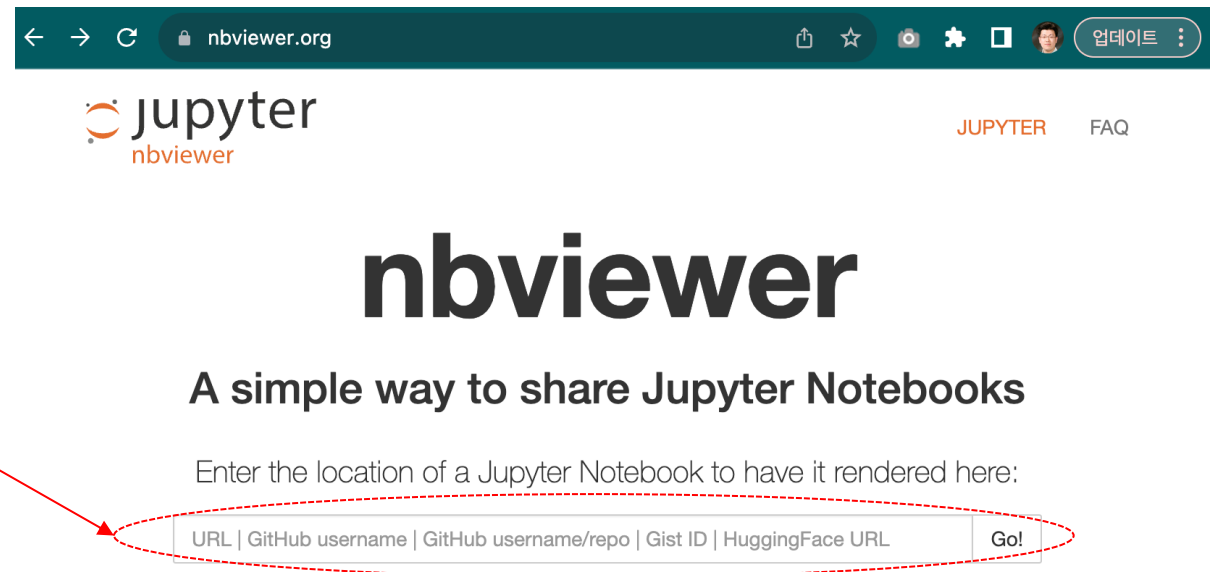
– 2) 업로드한 파일에 대한 URL 얻어오기

- 업로드한 파일을 Public Access 가능한지 점검 필요

• 예시: https://github.com/linklab/link_dl/blob/main/_02_homeworks/_01_tensor_handling/hw1_1.ipynb

– 3) 해당 URL을 다음 사이트의 텍스트 필드에 넣기

- <https://nbviewer.org/>



1. _02_Tensors - Self Practice

◆ _02_Tensors 폴더에서 제시하는 모든 예제를 ipynb에서 수행한 결과 제시

– 4) nbviewer 사이트에서 보여지는 자신의 숙제 확인하기

– 5) nbviewer 사이트에서 확인한 자신의 숙제 URL을 Copy 하기

- 예시: https://nbviewer.org/github/linklab/link_dl/blob/main/_02_homeworks/_01_tensor_handling/hw1_1.ipynb

– 6) Copy한 URL을 EL2 사이트 게시물 본문에 Paste 하여 넣기

– 7) 즉, 조교는 해당 사이트를 클릭하여 들어가 숙제를 확인할 예정임

nbviewer.org/github/linklab/link_dl/blob/main/_02_homeworks/_01_tens...

jupyter nbviewer

link_dl / _02_homeworks / _01_tensor_handling

a_tensor_initialization.py

```
In [2]: import torch
```

```
In [3]: # torch.Tensor class
t1 = torch.Tensor([1, 2, 3], device='cpu')
print(t1.dtype) # >>> torch.float32
print(t1.device) # >>> cpu
print(t1.requires_grad) # >>> False
print(t1.size()) # torch.Size([3])
print(t1.shape) # torch.Size([3])

torch.float32
cpu
False
torch.Size([3])
torch.Size([3])
```

```
In [4]: t2 = torch.tensor([1, 2, 3], device='cpu')
print(t2.dtype) # >>> torch.int64
print(t2.device) # >>> cpu
print(t2.requires_grad) # >>> False
print(t2.size()) # torch.Size([3])
print(t2.shape) # torch.Size([3])

torch.int64
cpu
False
torch.Size([3])
torch.Size([3])
```

• 위 코드에서 torch.tensor() 함수는 주어진 리스트내 원소 타입을 그대로 유지시켜 준다.

b_tensor_initialization_copy

```
In [ ]:
```

2. _03_real_world_data_to_tensors - Self Practice

◆ _03_real_world_data_to_tensors 폴더에서 제시하는 모든 예제를 ipynb에서 수행한 결과 제시

- 파일명: [hw1_2.ipynb](#)

```
▼ _03_real_world_data_to_tensors
  a_2d_image_data.py
  b_3d_image_data.py
  c_tabular_wine_data.py
  d_tabular_california_housing.py
  e_bikes_sharing_data.py
  f_hourly_bikes_sharing_data.py
  g_crypto_currency_data.py
  h_audio_data.py
  i_video_data.py
  j_linear_regression_dataset_dataloader.py
  k_2d_image_dataset_dataloader.py
  l_wine_dataset_dataloader.py
  m_california_housing_dataset_dataloader.py
  n_time_series_dataset_dataloader.py
  o_hourly_bikes_sharing_dataset_dataloader.py
  p_crypto_currency_dataset_dataloader.py
```

요구사항

◆ hw1_1.ipynb 및 hw1_2.ipynb에 내용 구성 요구사항

- 프로그램 코드를 실행되는 것을 확인할 수 있도록 출력값들을 정확하게 나타낼 것
 - 모든 코드에 대한 출력이 잘 나와 있는지 확인함
- 코드를 이해를 돕기 위하여 추가 코드 블록 삽입 가능
 - 주석을 넣어서 본 코드는 추가 코드임을 나타내어야 함
- 핵심 코드라고 생각되는 것들에 대해서 주석(한글 또는 영문)을 넣을 것
 - 다다익선
- 각 코드 블록에 대하여 자신이 취득한 기술적 사항/고찰 내용을 생각한 바를 제시할 것
 - 다다익선
- [NOTE] 숙제 마지막에는 "숙제 후기" 라는 섹션 제목으로 본 숙제를 한 이후의 느낀점, 하고싶은 말, 또는 불평 등을 반드시 넣을 것
 - "숙제 후기" 섹션이 없으면 감점 처리

요구사항

◆ 제출형태 및 방법

- Jupyter Notebook 파일명: [hw1_1.ipynb](#) 및 [hw1_2.ipynb](#)
- 숙제 제출 방식
 - <http://el2.koreatech.ac.kr> 의 “딥러닝및실습” 교과목 “과제” 메뉴
 - 숙제 게시물 제목: 딥러닝및실습-1차-홍길동-2020111222
 - 게시물 본문에는 앞 페이지에서 설명한 <https://nbviewer.org/> 로 시작하는 URL을 과제 1과 과제 2에 대하여 각각 넣기 (총 2개 URL 제시)
- 숙제에 관한 질의/응답
 - <http://el2.koreatech.ac.kr> 의 “딥러닝및실습” 교과목 “Q&A” 게시판 활용

◆ 제출기한

- 2024년 9월 24일 (화) 23시 59분
- 지각 제출은 받지 않습니다 (0점 처리)