

파이썬으로 배우는 딥러닝(Deep Learning)

0회차 수업
준비사항

강의에 관한 전반적인 안내

- ◆ 본 강의는 신경망의 기초부터 CNN, 전이학습, 암석식별머신실습으로 구성되어 있습니다.
- ◆ 본 강의는 수강생에게 딥러닝의 내부 작동 원리를 설명하기 위하여 조금은 어려운 수학(미분 등)이 등장합니다.
- ◆ 본 강의는 **딥러닝 내부 작동의 큰 흐름을 이해**하는 것을 목표로 하고 있습니다. 따라서, **세부적인 내용이 이해되지 않더라도** 크게 염려하시지 않아도 됩니다.
- ◆ 본 강의는 파이선 실습 코드 작성을 다수 포함하고 있으며, 딥러닝 이론 실습 외에도 파이선 자체의 코딩 연습의 의미도 포함하고 있습니다.
- ◆ 향후 머신러닝 프로젝트 수행을 위한 준비의 의미도 있어서 본 실습을 통해 **파이선 프로그래밍도 익숙해지는 시간이** 되기를 희망합니다.

퍼셉트론

신경망

신경망학습

오차역전파법

학습관련기술들

합성곱신경망

전이학습과 ResNet

암석식별머신실습

GitHub Codespaces에서 Python 개발

계정 생성

- <https://github.com>에서 계정 생성
- Education program이 존재해서 GitHub에서 제공하는 다양한 서비스를 큰 비용 부담 없이 사용가능 (GitHub Pro 무료 제공)



Codespaces 시작

- Visual Studio Code for the web이 GitHub에 연동
- 최근 소프트웨어 개발 시 이미 개발되어 있는 다양한 라이브러리 등을 활용하는 것이 필수적임
- 개발 초기에 이러한 요소를 구성하는 것이 생각보다 쉽지 않음
- Codespaces에서 제공하는 template 사용 (Python의 경우 Jupyter template이 제공됨)

Repository 생성

- Repository란 말 그대로 코드를 저장하는 저장소
- 새로운 프로젝트를 시작하기에 앞서 우선 repository를 만들고 관련된 모든 내용들을 같이 저장하면 관리가 용이함.
- 실습 시 편의상 Codespaces에서 template을 사용해서 만들어진 리소스를 repository에 저장하는 방식 사용

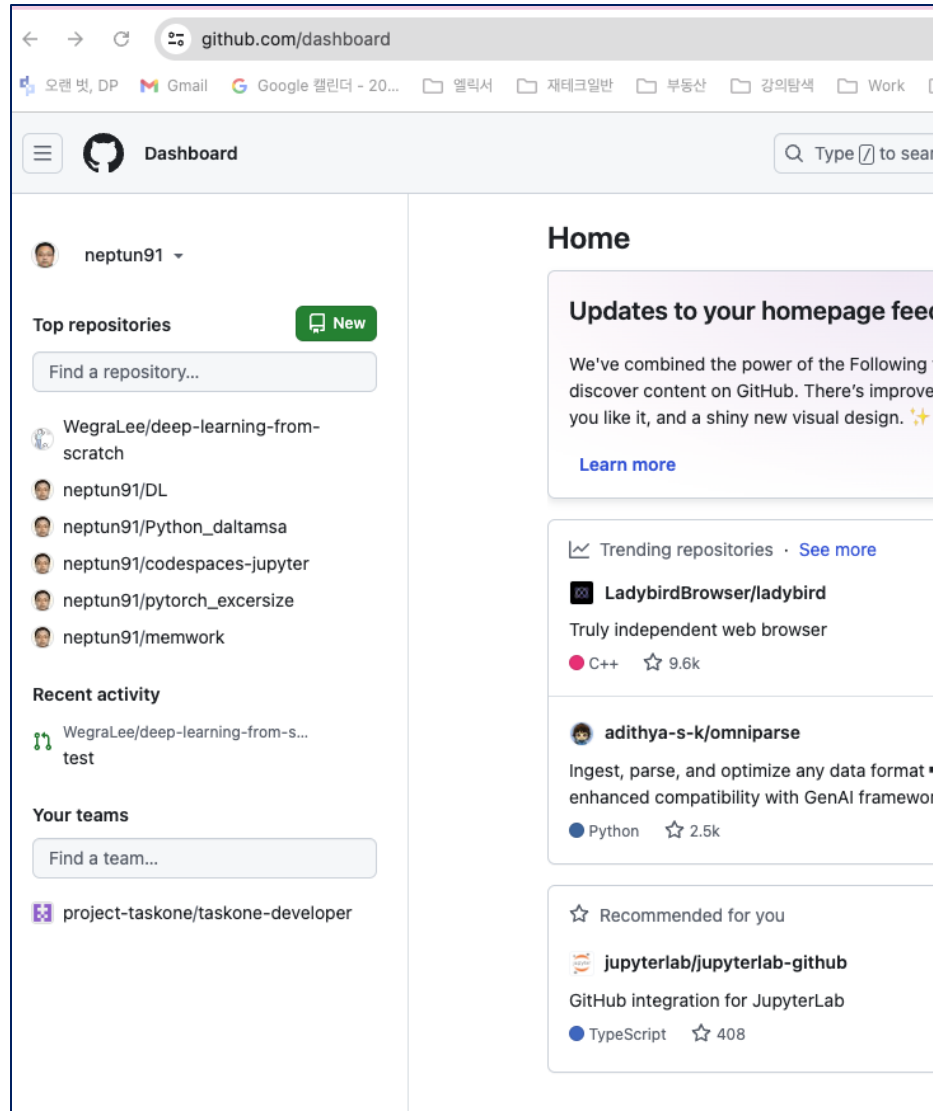
Codespaces 중지 및 재시동

- Codespaces가 실행되고 있는 동안(python code가 실행되지 않고 편집상태인 경우도 포함)에도 과금됨
- 사용이 끝난 경우 일시 정지시키면 비용 절약 가능

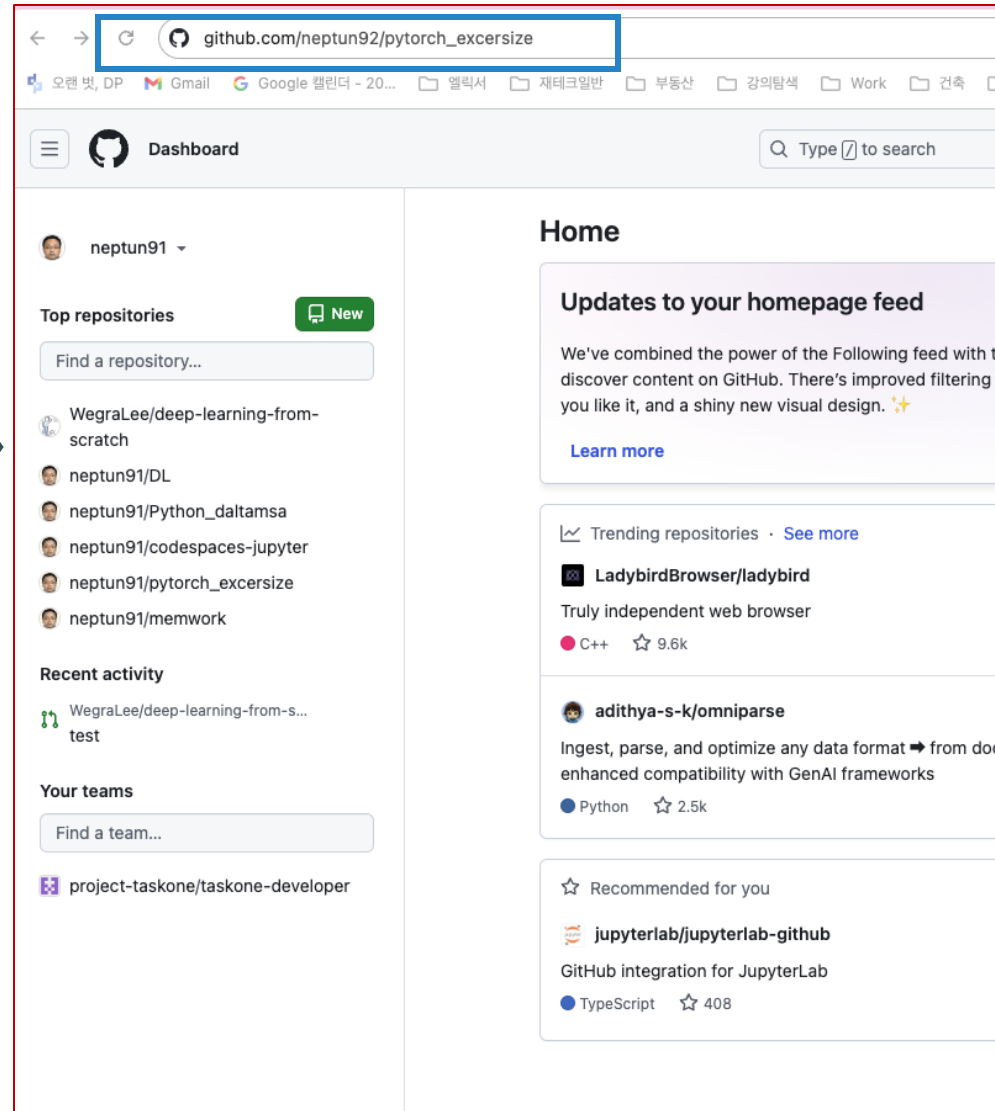
일반적인 workflow는 repository를 먼저 구성하지만, Codespaces의 template을 활용하기 위해 위와 같은 순서로 구성함

GitHub codespace 생성(1)

Step1. GitHub 로그인 상태

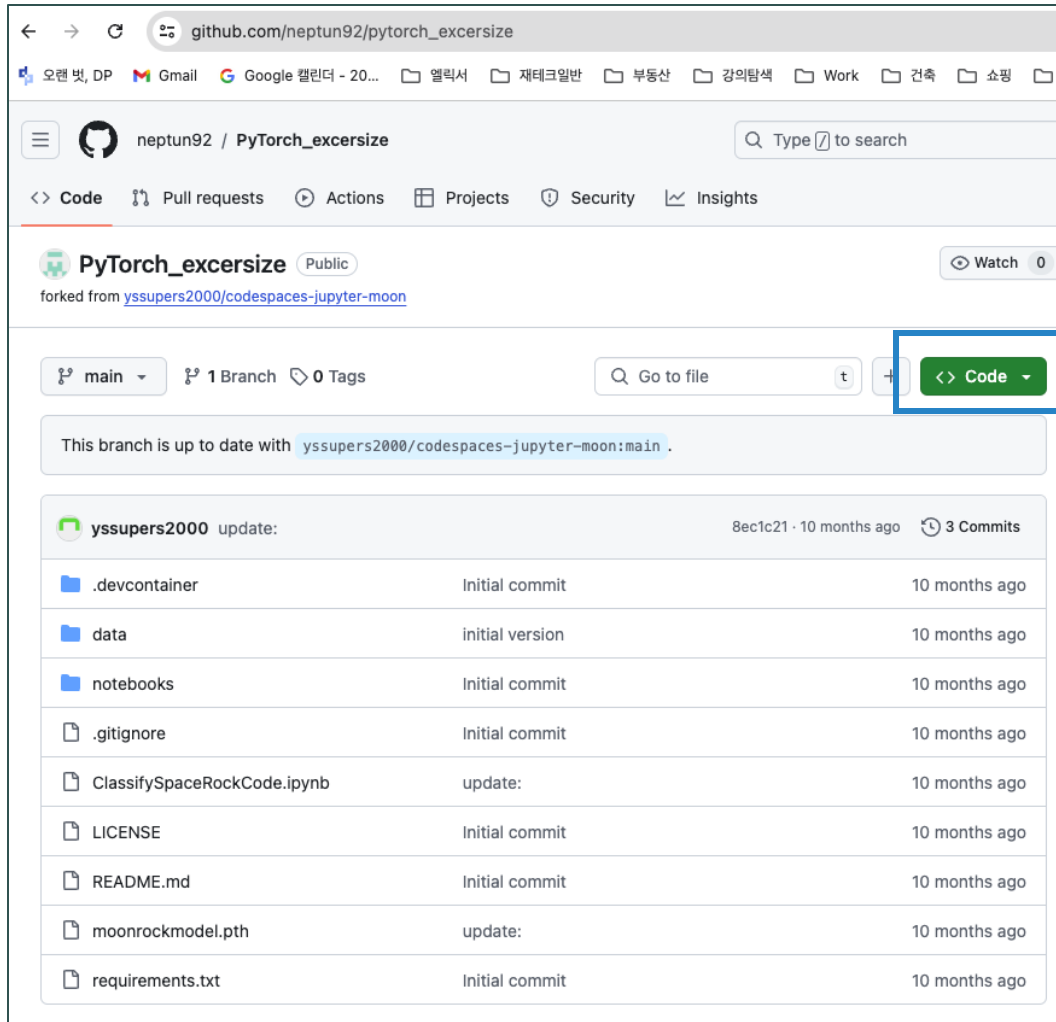


Step2 주소창입력 : <https://github.com/neptun92/DL-Excercise>

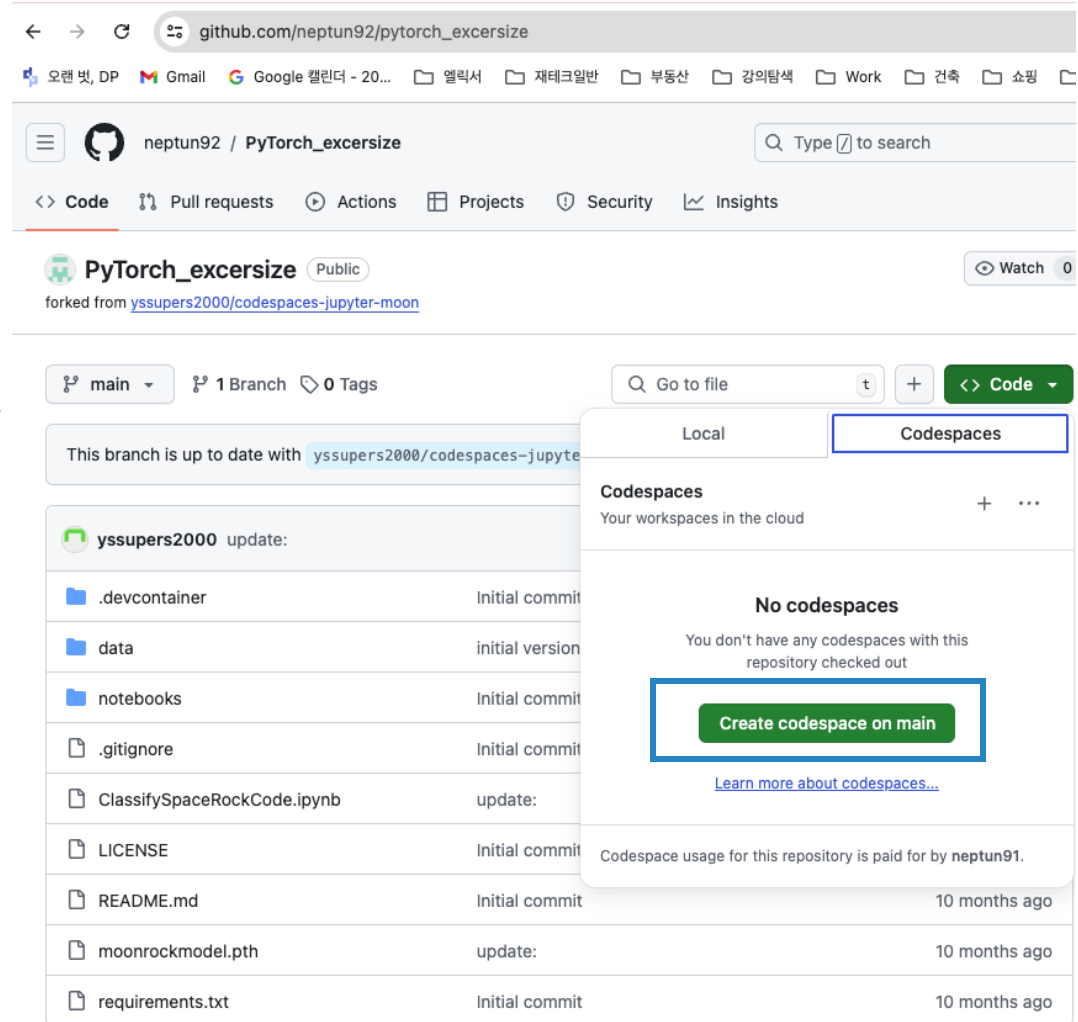


GitHub codespace 생성(2)

Step3. pytorch_excercise 소스 화면 → "<> Code" 버튼 클릭



Step4 'Create codespace on main' 클릭



GitHub codespace 생성(3)

Step5. codespace 생성 완료

실습파일

The screenshot displays a web browser window with the address bar showing `laughing-guacamole-7vr5rjqvvg65cpp9q.github.dev`. The browser's address bar and tabs are visible at the top. Below the browser window, a file explorer sidebar is open, showing a directory structure. The directory `DL3_20241006` is highlighted with a blue box. A blue arrow points from the text '실습파일' to this directory. The file explorer shows the following structure:

- DL-EXCCERSIZE [CODESPAC...
 - .devcontainer
 - data
 - notebooks
 - DL1_20240707
 - DL3_20241006 (highlighted)
 - ClassifySpaceRockCode.ipynb
 - image-classifier.ipynb
 - matplotlib.ipynb
 - population.ipynb
 - .gitignore
 - LICENSE
 - moonrockmodel.pth
 - README.md
 - requirements.txt

The main area of the browser shows the content of the `LICENSE` file, which is the MIT License. The text of the license is displayed in a monospace font, starting with 'MIT License' and 'Copyright (c) 2022 GitHub'.

GitHub codespace 생성(4)

Step6. DL3-20241006/ch01/and_gate.ipynb 파일 실행

The screenshot shows a GitHub Codespace environment. On the left, the file explorer displays a directory structure: `DL-EXCCERSIZE [CODESPAC...`, `.devcontainer`, `data`, `notebooks`, `DL1_20240707`, `DL3_20241006`, `ch00`, `ch01`, `__pycache__`, `.ipynb_aml_checkpoints`, `.amlnignore`, `.amlnignore.amltmp`, `and_gate.ipynb` (highlighted), `and_gate.ipynb.amltmp`, `nand_gate.ipynb`, `nand_gate.ipynb.amltmp`, `or_gate.ipynb`, `or_gate.ipynb.amltmp`, `README.md`, `xor_gate.ipynb`, `xor_gate.ipynb.amltmp`, `ch02`, `ch03`, `ch04`, `ch05`, `ch06`. A blue arrow points from the text "Open후 실행" to the `and_gate.ipynb` file. On the right, the Jupyter Notebook editor shows the code for `and_gate.ipynb`. The code defines an `AND` function and runs it for all combinations of `(0, 0)`, `(1, 0)`, `(0, 1)`, and `(1, 1)`. The output shows the results: `(0, 0) -> 0`, `(1, 0) -> 0`, `(0, 1) -> 0`, and `(1, 1) -> 1`.

```
# coding: utf-8
import numpy as np

def AND(x1, x2):
    x = np.array([x1, x2])
    w = np.array([0.5, 0.5])
    b = -0.7
    tmp = np.sum(w*x) + b
    if tmp <= 0:
        return 0
    else:
        return 1

if __name__ == '__main__':
    for xs in [(0, 0), (1, 0), (0, 1), (1, 1)]:
        y = AND(xs[0], xs[1])
        print(str(xs) + " -> " + str(y))
```

[2]

```
... (0, 0) -> 0
(1, 0) -> 0
(0, 1) -> 0
(1, 1) -> 1
```

Open후 실행