

DATA CREATOR CAMP

2024 데이터 크리에이터 캠프

대학부 실습영상

5강. 딥러닝 실습



과학기술정보통신부

NIA 지능정보원
한국지능정보사회진흥원

목차

1. Tensor Manipulation
2. DNN Modeling
3. CNN Modeling
4. VGGNet Modeling

✓ Tensor Manipulation

- 실습 링크: https://drive.google.com/file/d/1Fv9_k4NGzn_l1JEY8CQB7e5ieMRP7BG0/view?usp=sharing

The screenshot shows a Jupyter Notebook interface with a table of contents on the left and code cells on the right. The table of contents includes sections like 'Tensor 기초', 'Tensor 조작', and various operations. The code cells demonstrate creating and manipulating tensors.

```
[1] import numpy as np
import matplotlib.pyplot as plt

import torch
import torch.nn as nn # 딥러닝, 즉 인공 신경망 모델을 설계할 때 필요한 함수를 모아 놓은 모듈
import torch.nn.functional as F # 그 중 자주 사용되는 함수
```

Tensor 기초

```
[2] scalar1 = torch.tensor([1.]) # Scalar 생성
scalar2 = torch.tensor([3.]) # Scalar 생성

print(scalar1, scalar2)

tensor([1.]) tensor([3.])
```

```
[3] add_scalar = scalar1 + scalar2
print(add_scalar)

tensor([4.])
```

```
[4] sub_scalar = scalar1 - scalar2
print(sub_scalar)

tensor([-2.])
```

```
[5] mul_scalar = scalar1 * scalar2
print(mul_scalar)

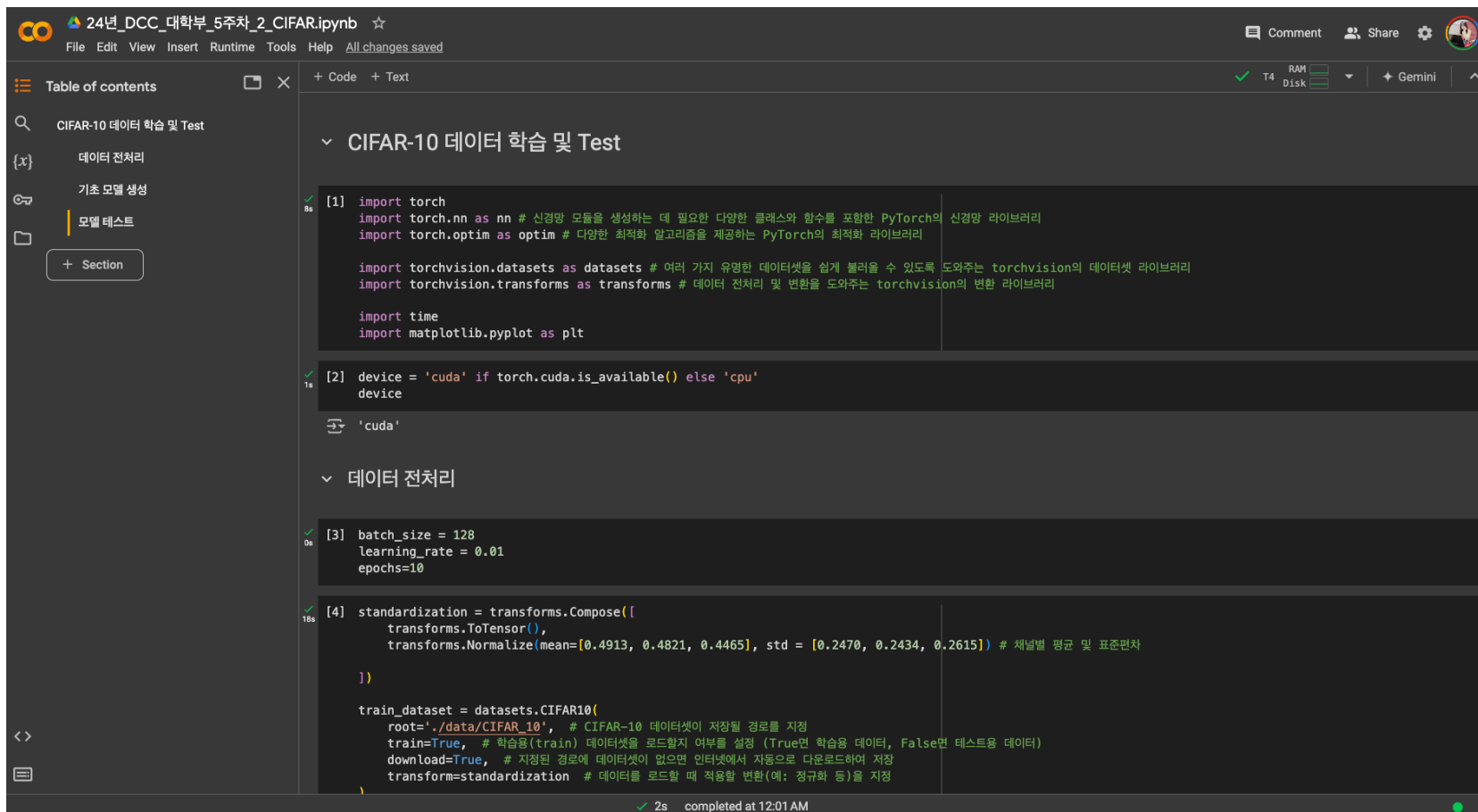
tensor([3.])
```

```
[6] div_scalar = scalar1 / scalar2
```

completed at 11:56 PM

✓ DNN Modeling

- 실습 링크: https://drive.google.com/file/d/1E88qBfAlg-Eq0IPrys_WS_v9ZenDwD70/view?usp=sharing



The screenshot displays a Jupyter Notebook interface with the following content:

- Table of contents:** CIFAR-10 데이터 학습 및 Test, 데이터 전처리, 기초 모델 생성, 모델 테스트.
- Section: CIFAR-10 데이터 학습 및 Test**
 - [1] Imports:** Imports torch, torch.nn, torch.optim, torchvision.datasets, torchvision.transforms, time, and matplotlib.pyplot.
 - [2] Device:** Sets the device to 'cuda' if available, otherwise 'cpu'.
 - Section: 데이터 전처리**
 - [3] Hyperparameters:** batch_size = 128, learning_rate = 0.01, epochs = 10.
 - [4] Standardization:** Defines a standardization transform using torchvision.transforms.Compose, ToTensor, and Normalize.

✓ CNN Modeling

- 실습 링크: <https://drive.google.com/file/d/1AhRIqLOWFopdetlZxeLAvqRHSNWl3jZh/view?usp=sharing>

The screenshot shows a Jupyter Notebook interface with the following content:

- Table of contents:**
 - CNN 모델
 - 데이터 다운로드
 - 하이퍼파라미터 설정
 - 데이터로더 설정
 - 이미지 체크
 - CNN Modeling
 - Loss Function & Optimizer 설정
 - Train
 - 결과 시각화
- Code Cells:**
 - CNN 모델**
 - 활용 데이터셋: Dogs & Cats
 - 0. 데이터 다운로드**
 - Cell [2]: `!wget https://storage.googleapis.com/tensorflow-1-public/course2/cats_and_dogs_filtered.zip`
Output: `--2024-08-27 15:27:29-- https://storage.googleapis.com/tensorflow-1-public/course2/cats_and_dogs_filtered.zip`
`Resolving storage.googleapis.com (storage.googleapis.com)... 142.250.99.207, 74.125.20.207, 108.177.98.207, ...`
`Connecting to storage.googleapis.com (storage.googleapis.com)[142.250.99.207]:443... connected.`
`HTTP request sent, awaiting response... 200 OK`
`Length: 68606236 (65M) [application/zip]`
`Saving to: 'cats_and_dogs_filtered.zip'`

`cats_and_dogs_filte 100%[=====] 65.43M 103MB/s in 0.6s`
`2024-08-27 15:27:29 (103 MB/s) - 'cats_and_dogs_filtered.zip' saved [68606236/68606236]`
 - Cell [3]: `# zip 파일을 열고 압축된 파일을 data 폴더에 모두 추출`
`zip_ref = zipfile.ZipFile("../cats_and_dogs_filtered.zip", 'r')`

✓ VGGNet Modeling

- 실습 링크: <https://drive.google.com/file/d/1KU4z4FfVIO7FuKjpb1rCBSRvNJQvFAtH/view?usp=sharing>

```

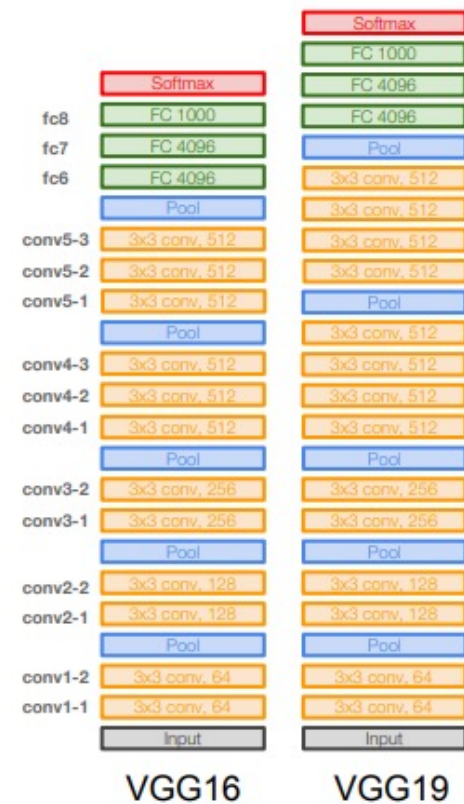
[1] import os
import numpy as np
import matplotlib.pyplot as plt

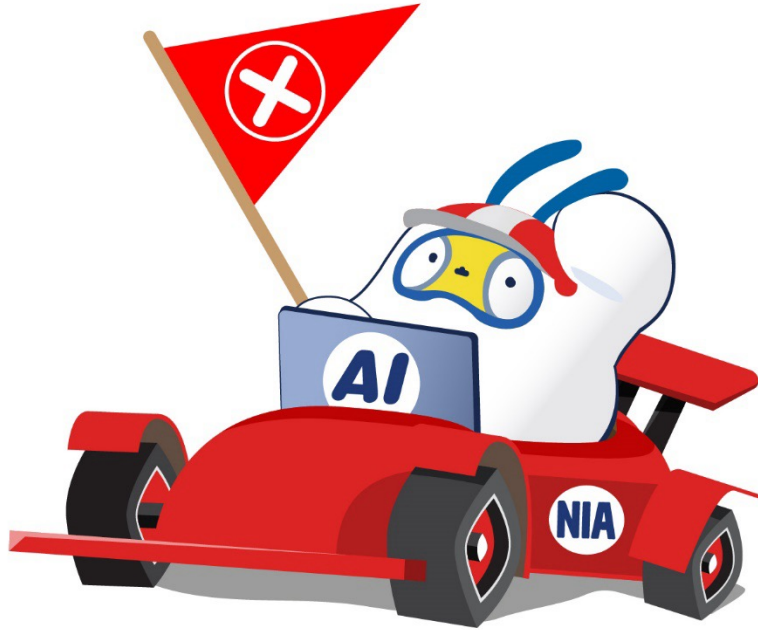
import torch
import torch.nn as nn
import torch.nn.functional as F
import torchvision
from torchvision import transforms, datasets

[2] batch_size = 32
epochs = 10
learning_rate = 0.001

[3] transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize((0.5,0.5,0.5),(0.5,0.5,0.5))
])

[4] train_dataset = datasets.CIFAR10(root = os.path.join('data', 'CIFAR_10'),
    train = True,
    download = True,
    transform = transform)
  
```





이 문서의
외부 유출 및
공유를 금합니다.

본 콘텐츠는 한국지능정보사회진흥원(NIA)의 동의 없이 무단 사용할 수 없으며,
상업적 목적으로 이용을 금합니다.

DATA CREATOR CAMP

2024 데이터 크리에이터 캠프

감사합니다.



과학기술정보통신부

NIA

지능정보원
한국지능정보사회진흥원