

딥러닝의 통계적이해

15강. 딥러닝 실습

1. 구글 Colaboratory
2. MNIST를 위한 신경망의 작성
3. CIFAR-10 신경망의 작성

한국방송통신대 이공희 교수

오늘의 **학습목표**

1. 구글 Colaboratory(코랩)를 이해한다.
2. Tensorflow-Keras를 이해한다.
3. 합성곱신경망을 작성하고 평가한다.

1. 구글 Colaboratory

1. 구글 Colaboratory

컴퓨팅 환경

- ◆ 딥러닝 모형 구현 컴퓨팅 환경 → 개별 PC에서는 복잡
 - GPU, Python, Tensorflow 등 딥러닝 프레임워크
 - 딥러닝 개발 환경 : 구글 Colaboratory(Colab)

1. 구글 Colaboratory

Colab의 개요

- ◆ 구글 Colab은 구글 클라우드의 가상 리눅스(Linux) 서버 기반 주피터 노트북(Jupyter notebook)
 - Python, Tensorflow, Keras, Numpy 등이 설치
 - GPU와 TPU 이용 가능
 - Python과 파일은 구글 드라이브에 저장, 이용되지 않는 중 가상머신이 중지되어 초기화

1. 구글 Colaboratory

Colab의 실습코드



<https://github.com/data-better/DeepS>

data-better / DeepS Unwa

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

master 1 branch 0 tags Go to file Add file Code

data-better Update 오타정리.md f063e25 11 hours ago 31 commits

| | | |
|----------------------------|----------------------|--------------|
| 10장_MNIST_DL.ipynb | Colaboratory를 통해 생성됨 | 10 days ago |
| Readme.md | Update Readme.md | 13 hours ago |
| 교재_10장_CIFAR_10_전이학습.ipynb | Colaboratory를 통해 생성됨 | 2 days ago |
| 오타정리.md | Update 오타정리.md | 11 hours ago |
| 제10장_CIFAR_10_DL.ipynb | Colaboratory를 통해 생성됨 | 2 days ago |
| 제10장_Colab.ipynb | Colaboratory를 통해 생성됨 | 10 days ago |
| 제10장_주성분_인자분석_오토인코더... | Colaboratory를 통해 생성됨 | 10 days ago |

1. 구글 Colaboratory

Colab의 실습코드



<https://github.com/data-better/DeepS>

 Open in Colab

Colab 소개

Colab의 수행

- Jupyter Notebook과 동일 : 코드와 텍스트로 구성
- 코드 셀에서 python 코드를 입력한 후 다음을 입력하여 수행
 - **Cmd/Ctrl+Enter** : 수행 후 커서가 셀에 남음
 - **Shift+Enter** : 수행 후 커서가 다음 셀로 이동
 - **Alt+Enter** : 수행 후 다음 코드 셀을 만들고 커서가 그 셀로 이동
 - 상위 메뉴 런타임에서 수행 관련 옵션을 지정

```
In [ ]: 1 + 0.1
```

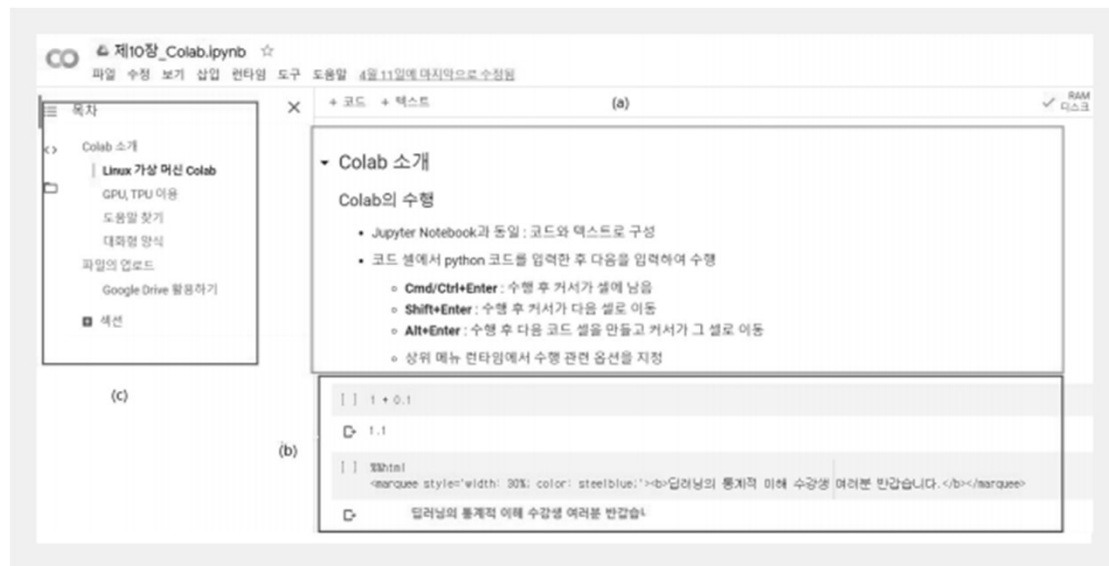
```
Out[ ]: 1.1
```

```
In [ ]: %%html
<marquee style='width: 30%; color: steelblue;'><b>딥러닝의 통계적 이해 수강생 여러
```

1. 구글 Colaboratory

Colab의 구조

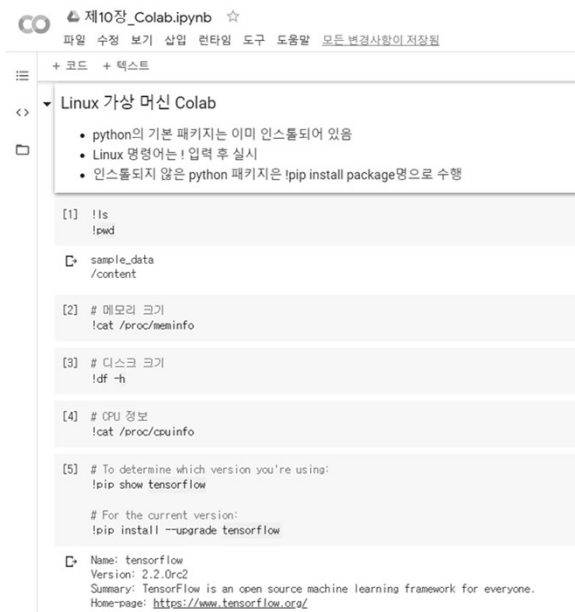
- ◆ Colab의 노트북은 코드와 텍스트로 나누어져 있음
 - (a) 텍스트 부분 (b) 코드 부분
 - (c) 수행 : Cmd/Ctrl + Enter, Shift + Enter, Alt+Enter



1. 구글 Colaboratory

Colab과 리눅스

- ◆ Colab은 우분투 리눅스(Linux) 가상머신
 - 리눅스 명령어 !를 입력 후 수행



```
CO 제10장_Colab.ipynb ☆
파일 수정 보기 삽입 런타임 도구 도움말 모든 변경사항이 저장됨

+ 코드 + 텍스트

Linux 가상 머신 Colab


- python의 기본 패키지는 이미インストール되어 있음
- Linux 명령어는 ! 입력 후 실시
- 인ストール되지 않은 python 패키지는 !pip install package명으로 수행



[1] !ls
!pwd

sample_data
/content

[2] # 메모리 크기
!cat /proc/meminfo

[3] # 디스크 크기
!df -h

[4] # CPU 정보
!cat /proc/cpuinfo

[5] # To determine which version you're using:
!pip show tensorflow

# For the current version:
!pip install --upgrade tensorflow

Name: tensorflow
Version: 2.2.0rc2
Summary: TensorFlow is an open source machine learning framework for everyone.
Home-page: https://www.tensorflow.org/
```

1. 구글 Colaboratory

Colab과 GPU

◆ 메뉴의 런타임 - 런타임 유형변경 GPU 또는 TPU로 설정

The screenshot shows the Google Colaboratory interface. At the top, the file name is '제10장_Colab.ipynb'. Below it, the '런타임' (Runtime) menu is highlighted. A dropdown menu is open, showing '3. GPU, TPU 이용' (Use GPU, TPU) and a sub-option '상단 메뉴 런타임 - 런타임유형변경에서 선택' (Select from the top menu Runtime - Change runtime type). Below this, a code cell is visible with the command `!nvidia-smi`. The output of the command is displayed as a table:

| NVIDIA-SMI 440.64.00 Driver Version: 418.67 CUDA | | | | | | | | | |
|--|----------|---------------|------------------|----------------|-----|--|--|--|--|
| GPU | Name | Persistence-M | Bus-Id | Disp.A | Vol | | | | |
| Fan | Temp | Perf | Pwr:Usage/Cap | Memory-Usage | GPU | | | | |
| 0 | Tesla P4 | Off | 00000000:00:04.0 | Off | | | | | |
| N/A | 34C | P8 | 7W / 75W | 0MIB / 7611MIB | | | | | |

Below the table, the 'Processes' section is shown with the message 'No running processes found'.

A '노트 설정' (Notebook Settings) dialog box is open over the code cell. It has a '런타임 유형' (Runtime type) dropdown set to 'Python 3'. Below it, the '하드웨어 가속기' (Hardware accelerator) dropdown is set to 'GPU'. A text box below the dropdown says: 'Colab을 최대한 활용하려면 필요하지 않은 경우 GPU를 사용하지 않는 것이 좋습니다. 자세히 알아보기' (To maximize Colab, it is recommended not to use GPU if not needed. Learn more). At the bottom of the dialog, there is a checkbox '이 노트를 저장할 때 코드 셀 출력 생략' (Omit code cell output when saving this notebook) which is currently unchecked. The dialog has '취소' (Cancel) and '저장' (Save) buttons.

1. 구글 Colaboratory

Colab과 파일관리

◆ 로컬 PC 또는 구글 드라이브와 연결해서 관리

The screenshot displays the Google Colaboratory interface. On the left, the 'Files' sidebar shows a file named 'flower2.jpg' under the 'sample_data' folder, labeled (b). The main workspace shows a Jupyter notebook titled '제10장_Colab.ipynb'. The notebook content includes a code cell with the following Python code:

```
[1] from google.colab import files
uploaded = files.upload()

for fn in uploaded.keys():
    print('User uploaded file "(name)" with length (length) bytes'.format(
        name=fn, length=len(uploaded[fn])))
```

Below the code cell, the output shows the file upload process, labeled (a):

```
File [flower2.jpg] is uploading...
• flower2.jpg(image/jpeg) - 1269128 bytes, last modified: 2020. 3. 3. - 100% done
Saving flower2.jpg to flower2.jpg
User uploaded file "flower2.jpg" with length 1269128 bytes
```

Below the output, the 'Google Drive 활용하기' section is visible, labeled (c). It includes a link to connect to Google Drive and a code cell with the following Python code:

```
# Google drive 연결
from google.colab import drive
drive.mount('/content/drive')
```

The output of the code cell shows the URL to go to in a browser and the authorization code, labeled (c):

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=9473189898
Enter your authorization code:
Mounted at /content/drive
```

1. 구글 Colaboratory

Colab 실습

◆ Colab 실습

The screenshot displays the Google Colaboratory web interface. At the top, the notebook is titled '제10장_Colab.ipynb'. Below the title, there are tabs for '파일' (File), '수정' (Edit), '보기' (View), '삽입' (Insert), '런타임' (Runtime), '도구' (Tools), and '도움말' (Help). The '도움말' tab is currently selected, showing a sidebar with a '목차' (Table of Contents) and a main content area. The sidebar lists various topics under 'Colab 소개' (Colab Introduction), including 'Linux 가상 머신 Colab', 'GPU, TPU 이용', '도움말 찾기', '대화형 양식', '파일의 업로드', 'Google Drive 활용하기', and '섹션'. The main content area is titled 'Colab 소개' and 'Colab의 수행' (Colab Execution). It contains a list of bullet points explaining how to execute code in Colab, such as using 'Cmd/Ctrl+Enter' to run the current cell, 'Shift+Enter' to run and move to the next cell, and 'Alt+Enter' to run and create a new cell below. Below the text, there is a code input field with the text '1 + 0.1', which has been executed, resulting in the output '1.1'. At the bottom, there is a section of HTML code that creates a marquee effect, displaying the text '답러님의 통계적 이해 수강생 여러분' (Dear students of the Statistical Understanding course).

2. MNIST를 위한 신경망의 작성

2. MNIST를 위한 신경망의 작성

MNIST 데이터

- ◆ 숫자 손글씨 이미지 데이터인 MNIST
 - 60,000개의 28x28 흑백 이미지 훈련데이터와 10,000개의 시험 데이터
- ◆ Tensorflow 2.0의 Keras(tf.keras)를 이용
 - 완전연결 신경망과 합성곱 신경망

2. MNIST를 위한 신경망의 작성

완전연결 신경망

- ◆ MNIST 데이터를 이용하여 손글씨를 식별하는 은닉층 1개의 완전연결 신경망

```
1 import tensorflow as tf
2 from tensorflow.keras import datasets, models
3 from tensorflow.keras.layers import Flatten,
  Dense, Dropout, Conv2D, MaxPool2D
4 from tensorflow.keras.utils import plot_model
5 print(tf.__version__)
```

2. MNIST를 위한 신경망의 작성

완전연결 신경망

```
6 mnist = datasets.mnist
7 (train_x, train_y), (test_x, test_y)=mnist.load_data()
8 train_x, test_x = train_x / 255.0, test_x / 255.0
9 import matplotlib.pyplot as plt
10 for col1 in range(16):
11     plt.subplot(4,4,col1+1)
12     plt.imshow(train_x[col1].reshape(28,28),
13               cmap=plt.cm.binary)
13 plt.show()
```


2. MNIST를 위한 신경망의 작성

완전연결 신경망

```
14 digit = train_x[0]
15 print("digit :", digit.shape)
16 print("train images :", train_x.shape)
17 print("test images :", test_x.shape)
```

2. MNIST를 위한 신경망의 작성

완전연결 신경망

```
18 model1 = models.Sequential([
19     Flatten(input_shape=(28, 28)),
20     Dense(512, activation='relu'),
21     Dense(10, activation='softmax')
22 ])
23 model1.summary()
24 plot_model(model1, to_file="model1_mnist.png",
    show_shapes=True)
```

2. MNIST를 위한 신경망의 작성

완전연결 신경망

| | |
|----|---|
| 25 | <code>model1.compile(optimizer='adam',</code> |
| | <code>loss='sparse_categorical_crossentropy',</code> |
| 26 | <code>metrics=['accuracy'])</code> |
| 27 | <code>hist = model1.fit(train_x, train_y, epochs=12,</code> |
| 28 | <code>batch_size=256, validation_split=0.25)</code> |

2. MNIST를 위한 신경망의 작성

완전연결 신경망

```
29 plt.plot(hist.history['accuracy'], 'b-')
30 plt.plot(hist.history['val_accuracy'], 'r--')
31 plt.legend(['train', 'test'], loc='upper left')
32 plt.ylim([0.94, 1.005])
33 plt.xlabel('Epoch')
34 plt.ylabel('Accuracy')
35 plt.show()
36 sc = model1.evaluate(test_x, test_y)
37 print("accuracy : ", sc[1], "   loss : ", sc[0])
```

2. MNIST를 위한 신경망의 작성

Colab 실습

◆ Colab 실습

10장_MNIST_DL.ipynb
파일 수정 보기 삽입 런타임 도구 도움말

목차

- MNIST 숫자 손글씨 분류
 - MNIST 데이터를 읽고 그래프로 살펴보기
 - 데이터 확인
 - 완전연결 신경망의 작성
 - 합성곱 신경망
- 섹션

MNIST 숫자 손글씨 분류

- Tensorflow 버전 검토

```
[ ] import tensorflow as tf
    from tensorflow.keras import datasets, models
    from tensorflow.keras.layers import Flatten, Dense, Dropout, Conv2D, MaxPool2D
    from tensorflow.keras.utils import plot_model
    print(tf.__version__)
```

2.2.0-rc2

MNIST 데이터를 읽고 그래프로 살펴보기

```
[ ] mnist = datasets.mnist

(train_x, train_y), (test_x, test_y) = mnist.load_data()
train_x, test_x = train_x / 255.0, test_x / 255.0

import matplotlib.pyplot as plt
for col1 in range(16):
    plt.subplot(4,4,col1+1)
    plt.imshow(train_x[col1].reshape(28,28), cmap=plt.cm.binary)
plt.show()
```

Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz>
11493376/11490434 [=====] - 0s 0us/step

2. MNIST를 위한 신경망의 작성

합성곱 신경망

- ◆ 2차원의 MNIST의 훈련 데이터와 시험데이터를 3차원 28x28x1로 전환

| | |
|----|--|
| 38 | <code>train_x = train_x.reshape(-1,28,28,1)</code> |
| 39 | <code>test_x = test_x.reshape(-1,28,28,1)</code> |

2. MNIST를 위한 신경망의 작성

합성곱 신경망

```
40 model2 = models.Sequential([
41     Conv2D(input_shape=(28,28,1), kernel_size=(3,3),
42           filters=32),
43     MaxPool2D(),
44     Conv2D(kernel_size=(3,3), filters=64),
45     MaxPool2D(),
46     Conv2D(kernel_size=(3,3), filters=128),
47     Flatten(),
48     Dense(512, activation='relu'),
49     Dropout(0.2),
50     Dense(10, activation='softmax')
51 ])
```

2. MNIST를 위한 신경망의 작성

합성곱 신경망

```
53 model2.compile(optimizer='adam',  
    loss='sparse_categorical_crossentropy',  
54    metrics=['accuracy'])  
55 hist2 = model2.fit(train_x, train_y, epochs=12,  
    batch_size=512,  
56    validation_split=0.25)
```


2. MNIST를 위한 신경망의 작성

Colab 실습

◆ Colab 실습

10장_MNIST_DL.ipynb
파일 수정 보기 삽입 런타임 도구 도움말

목차

- MNIST 숫자 손글씨 분류
 - MNIST 데이터를 읽고 그래프로 살펴보기
 - 데이터 확인
 - 완전연결 신경망의 작성
 - 합성곱 신경망
- 섹션

MNIST 숫자 손글씨 분류

- Tensorflow 버전 검토

```
[ ] import tensorflow as tf
    from tensorflow.keras import datasets, models
    from tensorflow.keras.layers import Flatten, Dense, Dropout, Conv2D, MaxPool2D
    from tensorflow.keras.utils import plot_model
    print(tf.__version__)
```

2.2.0-rc2

MNIST 데이터를 읽고 그래프로 살펴보기

```
[ ] mnist = datasets.mnist

(train_x, train_y), (test_x, test_y) = mnist.load_data()
train_x, test_x = train_x / 255.0, test_x / 255.0

import matplotlib.pyplot as plt
for col1 in range(16):
    plt.subplot(4,4,col1+1)
    plt.imshow(train_x[col1].reshape(28,28), cmap=plt.cm.binary)
plt.show()
```

Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz>
11493376/11490434 [=====] - 0s 0us/step

3. CIFAR-10 신경망의 작성

3. CIFAR-10 신경망의 작성

CIFAR-10 데이터

- ◆ 10개 분류군, 분류군당 6,000개의 32x32 저화질 컬러 이미지 데이터
 - 훈련데이터 5만개, 시험데이터 1만개

3. CIFAR-10 신경망의 작성

CIFAR-10 데이터 불러오기

```
1  cifar10 = datasets.cifar10
2  (train_x, train_y), (test_x, test_y) =
   cifar10.load_data()
9  train_x, test_x = train_x / 255.0, test_x / 255.0
```

3. CIFAR-10 신경망의 작성

합성곱 신경망의 작성

```
10 model3 = models.Sequential([
11     Conv2D(input_shape=(32,32,3), kernel_size=(3,3),
12         filters=32, activation="relu"),
13     MaxPool2D(),
14     Conv2D(kernel_size=(3,3), filters=64,
15         activation="relu"),
16     MaxPool2D(),
17     Conv2D(kernel_size=(3,3), filters=128,
18         activation="relu"),
19     Flatten(),
20     Dense(512, activation='relu'),
21     Dropout(0.2),
22     Dense(10, activation='softmax')
23 ])
```

3. CIFAR-10 신경망의 작성

합성곱 신경망의 학습

```
24 model3.compile(optimizer='adam',  
    loss='sparse_categorical_crossentropy',  
25    metrics=['accuracy'])  
26 hist3 = model3.fit(train_x, train_y, epochs=25,  
    batch_size=256,  
27    validation_split=0.25)  
36  
37 sc3 = model3.evaluate(test_x, test_y)  
38 print("accuracy : ", sc3[1], "    loss : ", sc3[0])
```

3. CIFAR-10 신경망의 작성

깊은 합성곱 신경망

```
39 model4 = models.Sequential([
40     Conv2D(input_shape=(32,32,3), kernel_size=(3,3),
padding='same',
41         filters=32, activation="relu"),
42     Conv2D(kernel_size=(3,3), filters=32,
activation="relu"),
43     MaxPool2D(),
44     Conv2D(kernel_size=(3,3), padding='same',
filters=64,
activation="relu"),
45     Conv2D(kernel_size=(3,3), filters=64,
activation="relu"),
46     MaxPool2D(),
```

3. CIFAR-10 신경망의 작성

깊은 합성곱 신경망

```
47 Conv2D(kernel_size=(3,3), padding='same',  
         filters=128, activation="relu"),  
48 Conv2D(kernel_size=(3,3), filters=128,  
         activation="relu"),  
49 Flatten(),  
50 Dense(512, activation='relu'),  
51 Dropout(0.5),  
52 Dense(256, activation='relu'),  
53 Dropout(0.5),  
54 Dense(10, activation='softmax')  
55 ])
```


3. CIFAR-10 신경망의 작성

깊은 합성곱 신경망의 학습

```
58 model4.compile(optimizer='adam',  
                  loss='sparse_categorical_crossentropy',  
59                  metrics=['accuracy'])  
60 hist4 = model4.fit(train_x, train_y, epochs=25,  
                    batch_size=256,  
61                    validation_split=0.25)  
62  
70 sc4 = model3.evaluate(test_x, test_y)  
71 print("accuracy : ", sc4[1], "   loss : ", sc4[0])
```

3. CIFAR-10 신경망의 작성

Colab 실습

◆ Colab 실습

제10장_CIFAR_10_DL.ipynb

파일 수정 보기 삽입 런타임 도구 도움말 변경사항을 저장할 수 없음

+ 코드 + 텍스트 드라이브로 복사

<> ▾ CIFAR-10 분류

- Tensorflow 버전 검토

```
[ ] import tensorflow as tf
    from tensorflow.keras import datasets, models, optimizers
    from tensorflow.keras.layers import Flatten, Dense, Dropout, Conv2D, MaxPool2D
    from tensorflow.keras.utils import plot_model
    import matplotlib.pyplot as plt
```

▾ CIFAR-10 데이터를 읽고 그래프로 살펴보기

```
[ ] cifar10 = datasets.cifar10
    (train_x, train_y), (test_x, test_y) = cifar10.load_data()
    print('train: x=%s, y=%s' % (train_x.shape, train_y.shape))
    print('test : x=%s, y=%s' % (test_x.shape, test_y.shape))
```

⚙ Downloading data from <https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz>
170500096/170498071 [=====] - 6s 0us/step
train: x=(50000, 32, 32, 3), y=(50000, 1)
test : x=(10000, 32, 32, 3), y=(10000, 1)

학습정리

- ✓ 구글 코랩(Colab)은 구글 클라우드로 제공되는 주피터 노트북(Jupyter notebook)이다.
- ✓ Tensorflow는 구글이 오픈소스로 공개한 머신러닝 라이브러리로 딥러닝 모형 작성에 가장 많이 이용되는 프레임워크이다.

학습정리

- ✓ Keras는 Tensorflow, MXNet, Theano 등을 백엔드로 이용할 수 있는 오픈 소스 딥러닝 프레임워크인데 Tensorflow와 결합되어 있다.

답러닝의 통계적이해
수고하셨습니다.

감사합니다.