



인공신경망과딥러닝심화

Lecture 02. 딥러닝의 핵심 미리 보기

동덕여자대학교
데이터사이언스 전공
권 범

목차

- ❖ 01. 미지의 일을 예측하는 원리
- ❖ 02. 딥러닝 코드 실행해 보기
- ❖ 03. 딥러닝 개괄하기
- ❖ 04. 이제부터가 진짜 딥러닝?

01. 미지의 일을 예측하는 원리

02. 딥러닝 코드 실행해 보기

03. 딥러닝 개괄하기

04. 이제부터가 진짜 딥러닝?

01. 미지의 일을 예측하는 원리

❖ 시작하기 전에 (1/2)



01. 미지의 일을 예측하는 원리

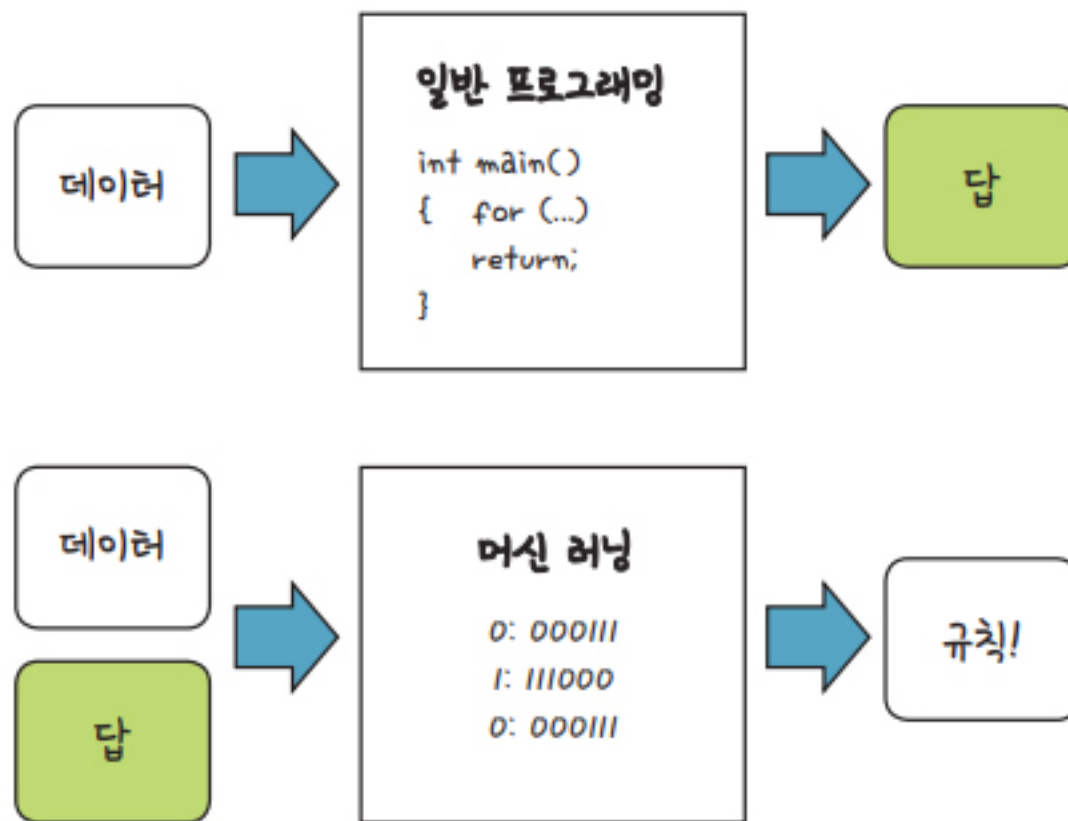
❖ 시작하기 전에 (2/2)

이러한 과제를 받았다고 가정

- ✓ 기존 프로그래밍 기법으로 이러한 프로그램을 만들려면 쉽지 않음
- ✓ 머신러닝은 이를 매우 쉽게 해결
- ✓ 기존에 우리가 했던 프로그래밍은 데이터를 입력해서 답을 구하는 데 초점이 맞추어 있음
- ✓ 머신러닝은 **데이터 안에서 규칙을 발견**하고 그 **규칙을 새로운 데이터에 적용해서 새로운 결과를 도출**하는 데 초점이 맞추어 있음
- ✓ 머신러닝은 기존 데이터를 이용해 아직 일어나지 않은 미지의 일을 예측하기 위해 만들어진 기법

01. 미지의 일을 예측하는 원리

❖ 머신러닝과 일반 프로그래밍 비교



01. 미지의 일을 예측하는 원리

❖ 머신러닝을 활용하는 방법 (1/5)

중환자를 전문으로 수술하는 어느 병원의 의사가 수많은 환자를 수술해 오던 중 다음과 같은 질문을 던져 보았음



“혹시 수술하기 전에 수술 후의 생존율을 수치로 예측할 수 있는 방법이 있을까?”

01. 미지의 일을 예측하는 원리

❖ 머신러닝을 활용하는 방법 (2/5)

자신이 그동안 집도한 수술 환자의 수술 전 상태와 수술 후 생존율을 정리해 놓은
데이터를 머신러닝 알고리즘에 넣는 것



머신러닝은 데이터가 가진 패턴과 규칙을 분석해서 저장해 둬



이후 새로운 환자가 오면 저장된 분석 결과와 비교해 생존 가능성을 예측

이것이 바로 머신러닝이 하는 일!

01. 미지의 일을 예측하는 원리

❖ 머신러닝을 활용하는 방법 (3/5)

- 여기서 데이터가 입력되고 패턴이 분석되는 과정을 **학습(Training)**이라고 함
- 학습 과정은 깨끗한 좌표 평면에 기존 환자들을 하나씩 배치하는 과정이라고 할 수 있음

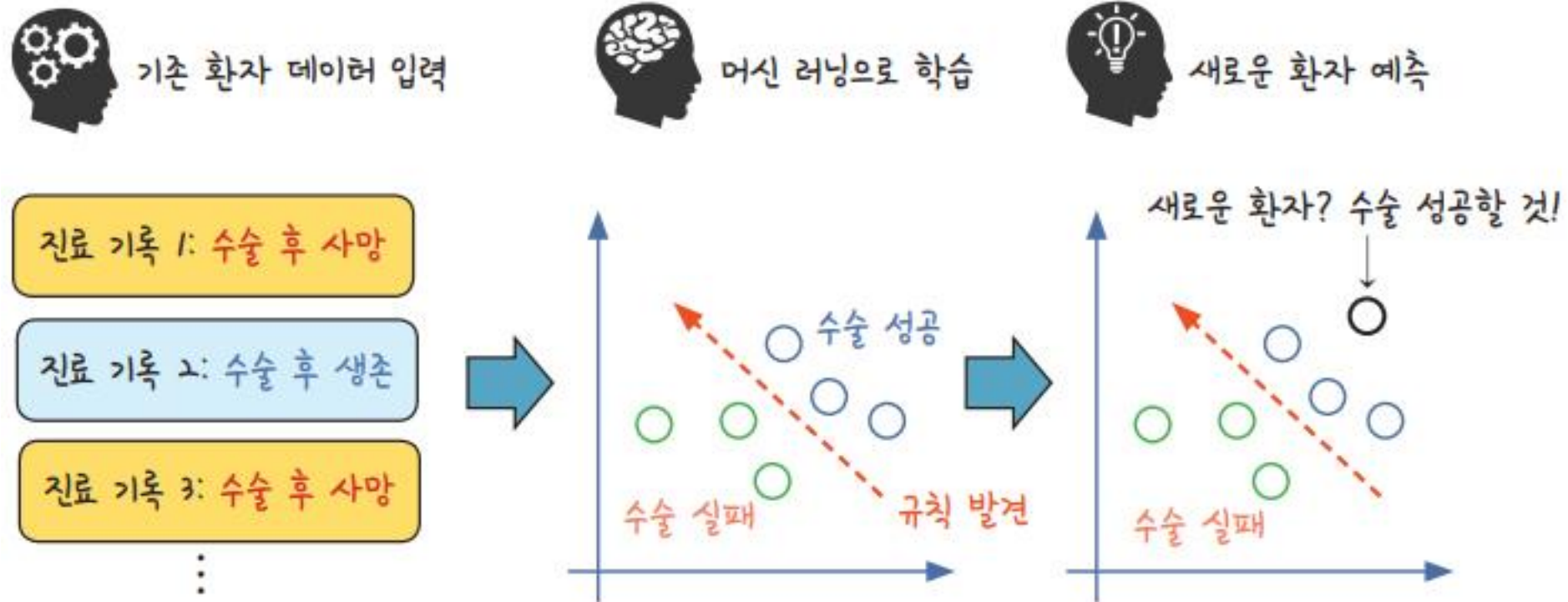
예를 들어

- ✓ 환자들의 분포를 그래프 위에 펼쳐 놓고 이 분포도 위에
수술 성공과 실패 여부를 구분짓는 경계를 그려 넣음
- ✓ 이를 잘 저장해 놓았다가 새로운 환자가 오면 분포도를 다시 꺼냄
- ✓ 새 환자가 분포도의 어디쯤 위치하는지 정하고는
아까 그려 둔 경계선을 기준으로 이 환자의 수술 결과를 예측하는 것

01. 미지의 일을 예측하는 원리

❖ 머신러닝을 활용하는 방법 (4/5)

머신러닝의 학습 및 예측 과정



01. 미지의 일을 예측하는 원리

❖ 머신러닝을 활용하는 방법 (5/5)

- 앞으로 수업시간에 배우려는 것이 바로 이러한 학습과 예측의 구체적인 과정
- 머신러닝의 예측 성공률은 결국 얼마나 정확한 경계선을 긋느냐에 달려 있음
- 더 정확한 선을 긋기 위한 여러 가지 노력이 계속되어 왔고,
그 결과 퍼셉트론(Perceptron), 아달라인(Adaline),
선형 회귀(Linear Regression) 등을 지나 오늘날 딥러닝이 탄생

02. 딥러닝 코드 실행해 보기

- 01. 미지의 일을 예측하는 원리
- 03. 딥러닝 개괄하기
- 04. 이제부터가 진짜 딥러닝?

02. 딥러닝 코드 실행해 보기

❖ 드라이브에 딥러닝 코드 저장하기 (1/2)

- 백문이 불여일견! 먼저 딥러닝의 코드를 불러와 그 형태를 살펴보고, 예측 결과가 나오는 과정을 미리 살펴볼 예정
- 딥러닝 소스 코드를 내 계정의 드라이브에 저장하고 실행하는 연습을 할 예정

“A picture is worth 1,000 words.”

02. 딥러닝 코드 실행해 보기

❖ 드라이브에 딥러닝 코드 저장하기 (2/2)

- 우선 'Lecture 02.ipynb'과 'ThoracicSurgery3.csv' 파일을 다운로드
- 그다음 나의 구글 계정 드라이브에 'Lecture 02.ipynb' 파일을 업로드

02. 딥러닝 코드 실행해 보기

❖ 딥러닝 코드 실행하기 (1/9)

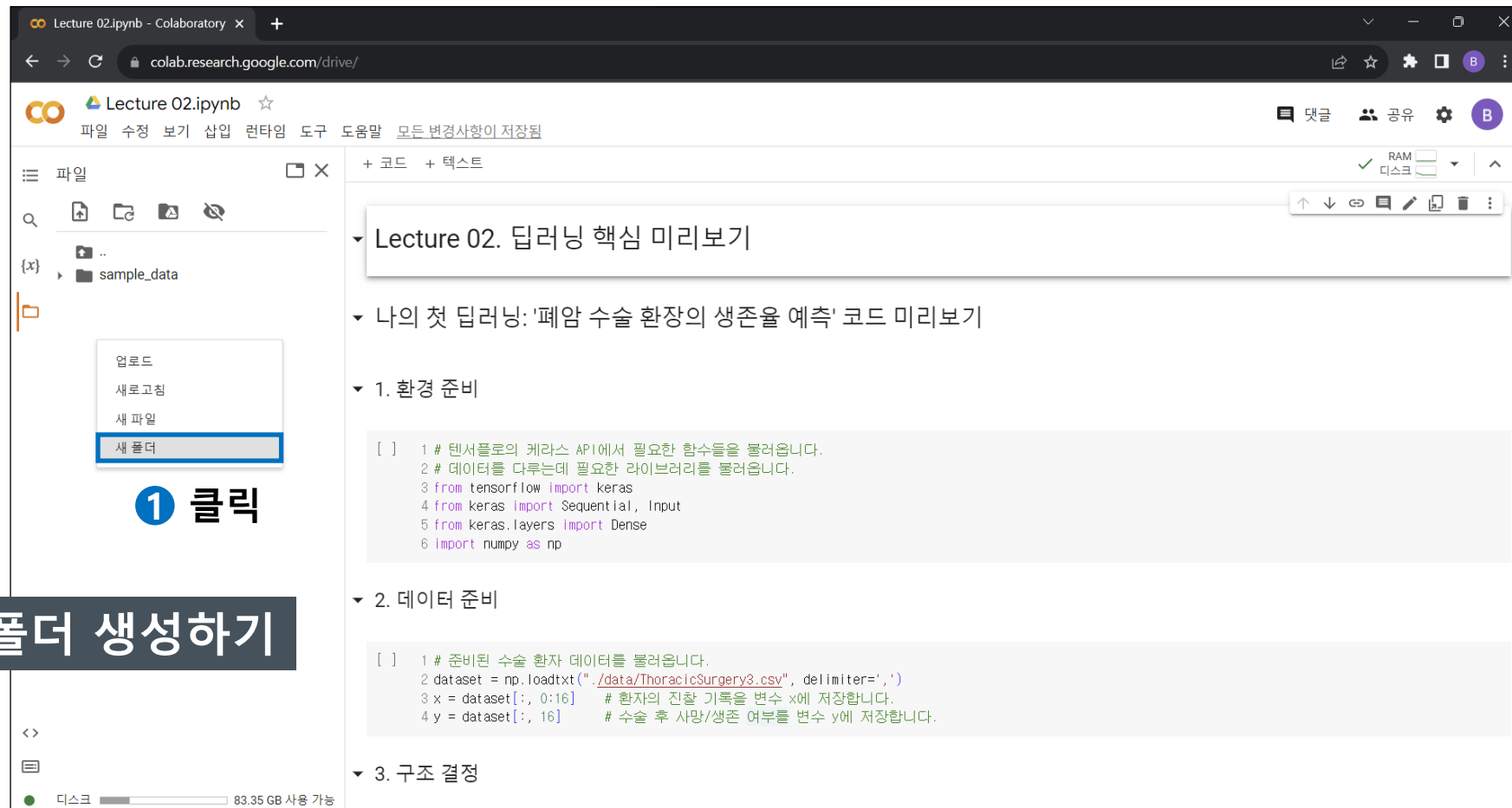
- 드라이브에 업로드한 'Lecture 02.ipynb' 파일을 더블클릭하면, 구글 코랩을 통해 파일이 열림
- 'ThoracicSurgery3.csv' 파일을 업로드하기 위해서 좌측의 ① 폴더를 클릭

구글 코랩으로 열기

02. 딥러닝 코드 실행해 보기

❖ 딥러닝 코드 실행하기 (2/9)

- 마우스의 오른쪽 버튼 클릭 > ① [새 폴더] 메뉴를 클릭해 새 폴더를 생성



02. 딥러닝 코드 실행해 보기

❖ 딥러닝 코드 실행하기 (3/9)

- 생성된 새 폴더의 이름을 'Untitled Folder'에서 'data'로 수정

폴더 이름 수정하기

The screenshot shows the Google Colaboratory web interface. On the left, the '파일' (Files) sidebar displays a folder named 'data' which is highlighted with a blue box. Below it, a 'sample_data' folder is visible. The main workspace area on the right contains a list of sections: 'Lecture 02. 딥러닝 핵심 미리보기', '나의 첫 딥러닝: '폐암 수술 환장의 생존율 예측' 코드 미리보기', '1. 환경 준비', '2. 데이터 준비', and '3. 구조 결정'. Under '1. 환경 준비', there is a code block with the following Python code:

```
[ ] 1 # 텐서플로의 케라스 API에서 필요한 함수들을 불러옵니다.  
2 # 데이터를 다루는데 필요한 라이브러리를 불러옵니다.  
3 from tensorflow import keras  
4 from keras import Sequential, Input  
5 from keras.layers import Dense  
6 import numpy as np
```

Under '2. 데이터 준비', there is another code block:

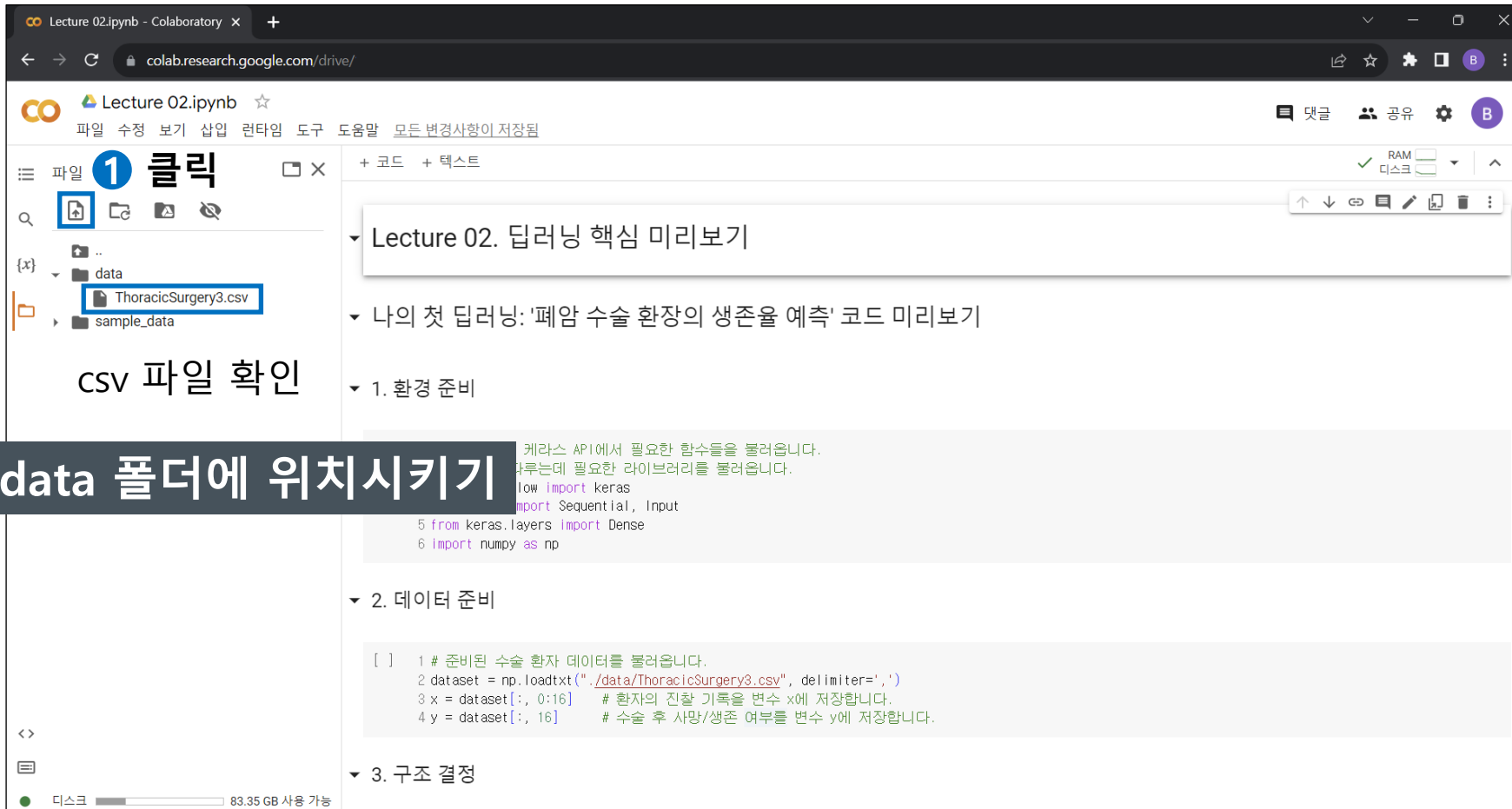
```
[ ] 1 # 준비된 수술 환자 데이터를 불러옵니다.  
2 dataset = np.loadtxt("./data/ThoracicSurgery3.csv", delimiter=',')  
3 x = dataset[:, 0:16] # 환자의 진찰 기록을 변수 x에 저장합니다.  
4 y = dataset[:, 16] # 수술 후 사망/생존 여부를 변수 y에 저장합니다.
```

The bottom of the interface shows a disk usage indicator: '디스크 83.35 GB 사용 가능'.

02. 딥러닝 코드 실행해 보기

❖ 딥러닝 코드 실행하기 (4/9)

- ① 업로드 아이콘  을 클릭하여 'ThoracicSurgery3.csv' 파일을 업로드하고, data 폴더로 옮김



The screenshot shows the Google Colaboratory interface. On the left, the file explorer shows a folder named 'data' containing a file named 'ThoracicSurgery3.csv'. A red box highlights the 'upload' icon (a square with an upward arrow) and the file 'ThoracicSurgery3.csv'. Below the file explorer, the text 'csv 파일 확인' is visible. In the center, a code cell is titled 'Lecture 02. 딥러닝 핵심 미리보기'. It contains a list of items: '나의 첫 딥러닝: '폐암 수술 환자의 생존율 예측' 코드 미리보기', '1. 환경 준비', '2. 데이터 준비', and '3. 구조 결정'. The code for '2. 데이터 준비' is expanded, showing the following Python code:

```
1 # 준비된 수술 환자 데이터를 불러옵니다.  
2 dataset = np.loadtxt("./data/ThoracicSurgery3.csv", delimiter=',')  
3 x = dataset[:, 0:16] # 환자의 진찰 기록을 변수 x에 저장합니다.  
4 y = dataset[:, 16] # 수술 후 사망/생존 여부를 변수 y에 저장합니다.
```

At the bottom of the code cell, the text 'csv 파일을 data 폴더에 위치시키기' is displayed in a dark box. The bottom status bar shows '디스크' and '83.35 GB 사용 가능'.

02. 딥러닝 코드 실행해 보기

❖ 딥러닝 코드 실행하기 (5/9)

- ① ✕아이콘을 클릭하면 목차 창을 닫을 수 있음 (또는 좌측 하단의 📁아이콘을 클릭해도 됨)

1 클릭

목차 창 닫기

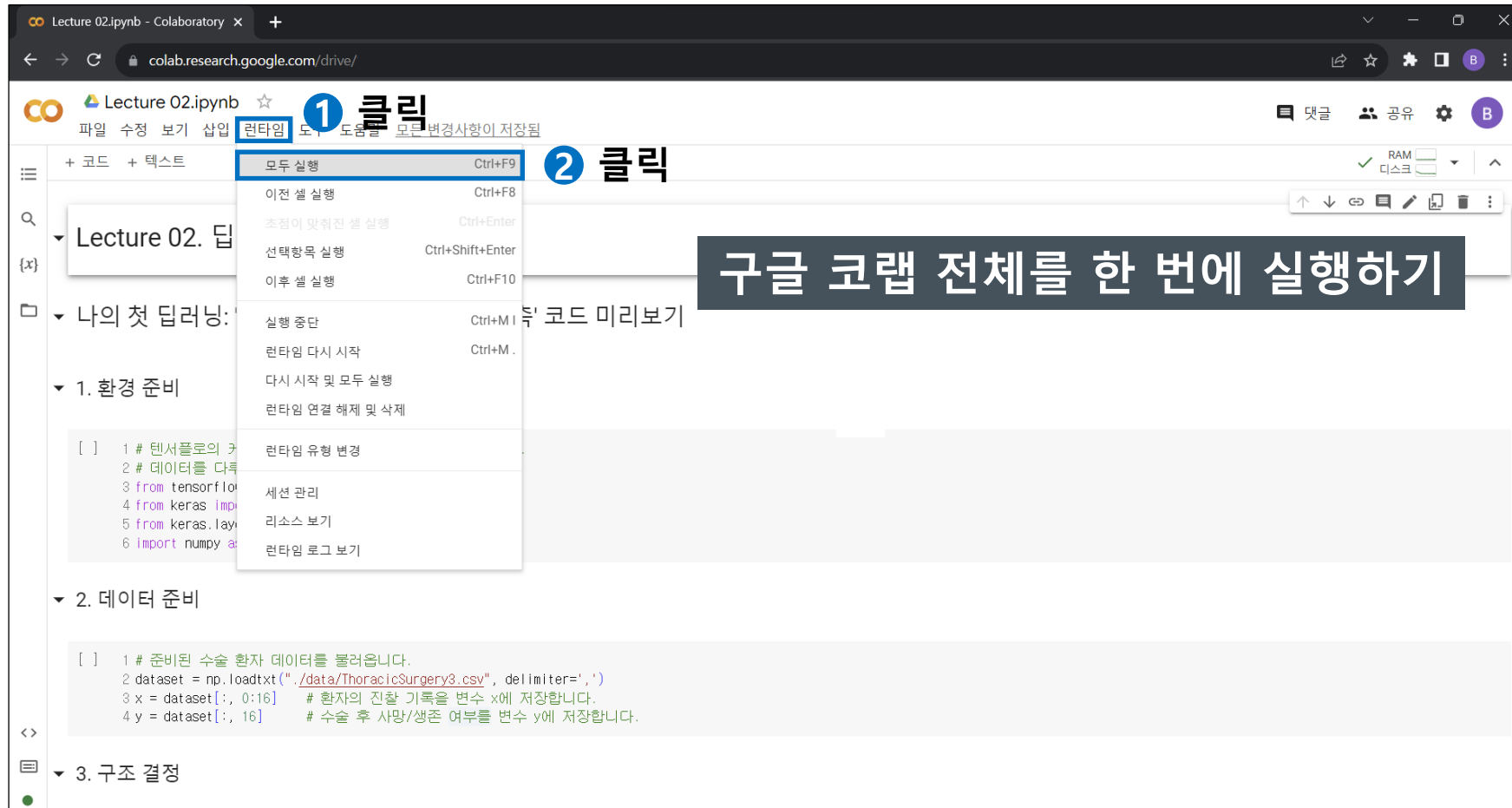
```
[ ] 1 # 텐서플로의 케라스 API에서 필요한 함수들을 불러옵니다.  
2 # 데이터를 다루는데 필요한 라이브러리를 불러옵니다.  
3 from tensorflow import keras  
4 from keras import Sequential, Input  
5 from keras.layers import Dense  
6 import numpy as np
```

```
[ ] 1 # 준비된 수술 환자 데이터를 불러옵니다.  
2 dataset = np.loadtxt("./data/ThoracicSurgery3.csv", delimiter=',')  
3 x = dataset[:, 0:16] # 환자의 진찰 기록을 변수 x에 저장합니다.  
4 y = dataset[:, 16] # 수술 후 사망/생존 여부를 변수 y에 저장합니다.
```

02. 딥러닝 코드 실행해 보기

❖ 딥러닝 코드 실행하기 (6/9)

- 구글 코랩 파일 전체를 한 번에 실행하려면 ① [런타임] > ② [모두 실행]을 선택



02. 딥러닝 코드 실행해 보기

❖ 딥러닝 코드 실행하기 (7/9)

- 코드별로 하나씩 실행하려면 각 코드창 앞의 실행 버튼 (▶)을 클릭하면 됨

1 클릭



```
1 # 텐서플로의 케라스 API에서 필요한 함수들을 불러옵니다.  
2 # 데이터를 다루는데 필요한 라이브러리를 불러옵니다.  
3 from tensorflow import keras  
4 from keras import Sequential, Input  
5 from keras.layers import Dense  
6 import numpy as np
```

⋮



4 클릭



```
1 # 딥러닝 모델을 실행합니다.  
2 model.compile(loss="binary_crossentropy",  
3               optimizer="adam",  
4               metrics=["accuracy"])  
5 history = model.fit(x, y, epochs=5, batch_size=16)
```

02. 딥러닝 코드 실행해 보기

❖ 딥러닝 코드 실행하기 (8/9)

- 코드창 맨 앞에 ,  아이콘이 차례로 나타나면서 코드가 실행되면 성공
- 실행을 마치면 다음과 같이 ① 각 창별 실행 시간이 나타나고 ② 실행 결과가 표시

① 실행 시간

```
1. 환경 준비
[1] 1 # 텐서플로의 케라스 API에서 필요한 함수들을 불러옵니다.
    2 # 데이터를 다루는데 필요한 라이브러리를 불러옵니다.
    3 from tensorflow import keras
    4 from keras import Sequential, Input
    5 from keras.layers import Dense
    6 import numpy as np

2. 데이터 준비
[2] 1 # 준비된 수술 환자 데이터를 불러옵니다.
    2 dataset = np.loadtxt("./data/ThoracicSurgery3.csv", delimiter=',')
    3 x = dataset[:, 0:16] # 환자의 진찰 기록을 변수 x에 저장합니다.
    4 y = dataset[:, 16]  # 수술 후 사망/생존 여부를 변수 y에 저장합니다.

3. 구조 결정
[3] 1 # 딥러닝 모델의 구조를 결정합니다.
    2 model = Sequential()
    3 model.add(Input(shape=(16,)))
    4 model.add(Dense(30, activation="relu"))
    5 model.add(Dense(1, activation="sigmoid"))

4. 모델 실행
[4] 1 # 딥러닝 모델을 실행합니다.
    2 model.compile(loss="binary_crossentropy",
    3               optimizer="adam",
    4               metrics=["accuracy"])
    5 history = model.fit(x, y, epochs=5, batch_size=16)

Epoch 1/5
30/30 [=====] - 1s 2ms/step - loss: 1.8923 - accuracy: 0.4447
Epoch 2/5
30/30 [=====] - 0s 2ms/step - loss: 0.5071 - accuracy: 0.8511
Epoch 3/5
30/30 [=====] - 0s 2ms/step - loss: 0.4552 - accuracy: 0.8511
Epoch 4/5
30/30 [=====] - 0s 2ms/step - loss: 0.4454 - accuracy: 0.8511
Epoch 5/5
30/30 [=====] - 0s 2ms/step - loss: 0.4400 - accuracy: 0.8511
```

구글 코랩 실행 결과

② 실행 결과

02. 딥러닝 코드 실행해 보기

❖ 딥러닝 코드 실행하기 (9/9)

- 실행 결과는 매번 실행할 때마다 미세하게 달라짐
- 이것은 첫 가중치를 랜덤하게 정하고 실행을 반복하며, 조금씩 가중치를 수정해 가는 딥러닝의 특성 때문임
- 딥러닝의 동작 원리에 대해서 앞으로 차차 배워 나갈 것

03. 딥러닝 개괄하기

- 01. 미지의 일을 예측하는 원리
- 02. 딥러닝 코드 실행해 보기
- 04. 이제부터가 진짜 딥러닝?

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰 (1/4)

- 지금 불러와 실행한 코드는 폐암 수술 환자의 수술 1년 후 생존율을 예측한 모델
- 먼저 코드를 개괄적으로 살펴보며 딥러닝을 프로그래밍하는 과정에 대한 감을 잡아 볼 예정
- 단 몇 줄로 이루어진 간략한 코드는 다음과 같이 크게 네 부분으로 나뉘어 있음

1. 환경 준비

딥러닝을 구동하는 데 필요한 라이브러리 호출

2. 데이터 준비

데이터를 불러와 사용할 수 있도록 준비

3. 구조 결정

어떤 딥러닝 구조를 만들 것인가

4. 모델 실행

만든 딥러닝을 실행시키고 결과 확인

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰 (2/4)

1. 환경 준비

```
from tensorflow import keras
from keras import Sequential, Input
from keras.layers import Dense
import numpy as np
```

딥러닝을 구동하거나 데이터를 다루는 데 필요한 라이브러리들을 불러옵니다.

2. 데이터 준비

```
dataset = np.loadtxt("./data/ThoracicSurgery3.csv", delimiter=',')
x = dataset[:, 0:16]
y = dataset[:, 16]
```

- ✓ 준비된 수술 환장 정보 데이터를 나의 구글 코랩 계정에 저장합니다.
- ✓ 해당 파일을 불러와 환자 상태의 기록에 해당하는 부분을 변수 x에, 수술 1년 후 사망/생존 여부를 변수 y에 저장합니다.

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰 (3/4)

3. 구조 결정

```
model = Sequential()  
model.add(Input(shape=(16,)))  
model.add(Dense(30, activation="relu"))  
model.add(Dense(1, activation="sigmoid"))
```

- ✓ 딥러닝 모델의 구조를 결정합니다.
- ✓ 여기에 설정된 대로 딥러닝을 수행합니다.

4. 모델 실행

```
model.compile(loss="binary_crossentropy",  
              optimizer="adam",  
              metrics=["accuracy"])  
history = model.fit(x, y, epochs=5, batch_size=16)
```

- ✓ 딥러닝 모델을 실행합니다.
- ✓ 앞서 설정된 구조대로 실행하고 결과를 출력합니다.

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰 (4/4)

- 우리가 다룰 모든 코드는 **파이썬**으로 되어 있음
- 파이썬은 초보자부터 전문가까지 모두에게 애용되는 프로그래밍 언어로,
특히 다양한 플랫폼에서 데이터를 분석하고 딥러닝, 머신러닝을 구현하는 데 사용
- 파이썬은 풍부한 라이브러리를 가지고 있다는 것이 장점인데,
라이브러리란 특정한 기능을 담은 작은 프로그램들(Module, API)을 모아 놓은 것을 의미
- 목적에 따라 라이브러리를 불러오면 다양한 작업을 간단히 진행할 수 있음
- 라이브러리를 불러올 때 사용하는 명령어가 import

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ① 환경 준비 (1/3)

- 코드의 처음이 다음과 같이 시작

```
from tensorflow import keras  
from keras import Sequential, Input  
from keras.layers import Dense  
import numpy as np
```

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ① 환경 준비 (2/3)

- 라이브러리에 포함된 모듈이 너무 많을 때,
그중 지금 필요한 일부 모듈만 다음과 같이 불러올 수 있음

```
from (라이브러리명) import (함수명)
```

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ① 환경 준비 (3/3)

- 예를 들어, **from** keras **import** Sequential, Input은 케라스(Keras)라는 API로부터 Sequential() 함수와 Input() 함수를 불러오라는 의미
- 마찬가지로, **from** keras.layers **import** Dense는 케라스 API의 레이어(Layers) 클래스로부터 Dense()라는 함수를 불러오라는 의미
- 불러온 라이브러리명이 길거나 같은 이름이 이미 있을 경우 다음과 같이 짧게 줄일 수도 있음

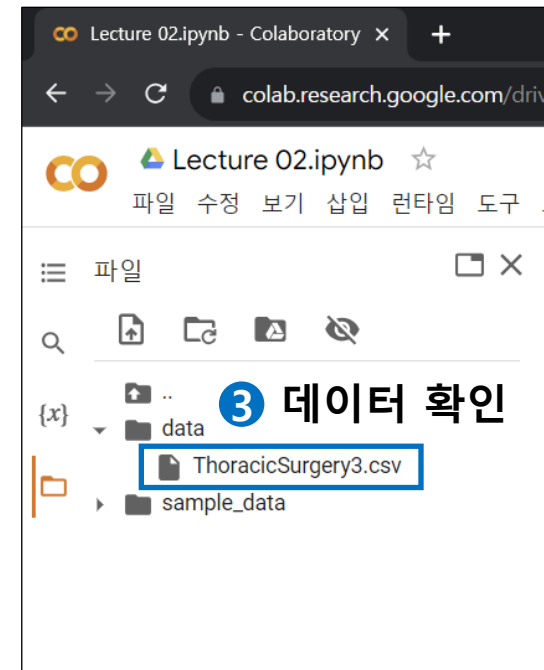
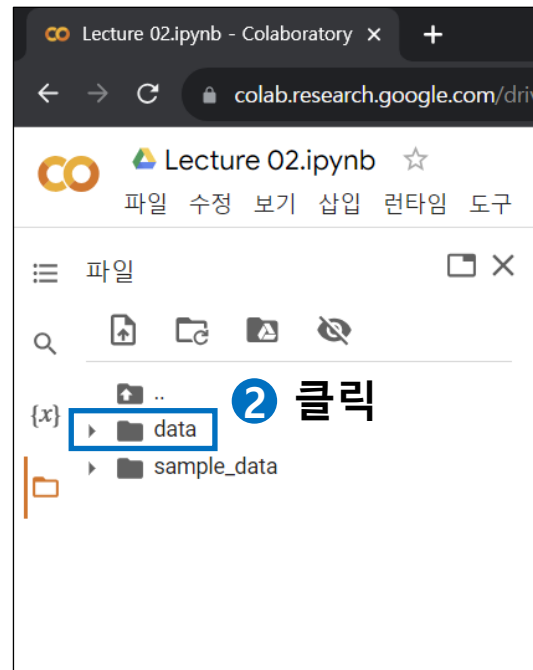
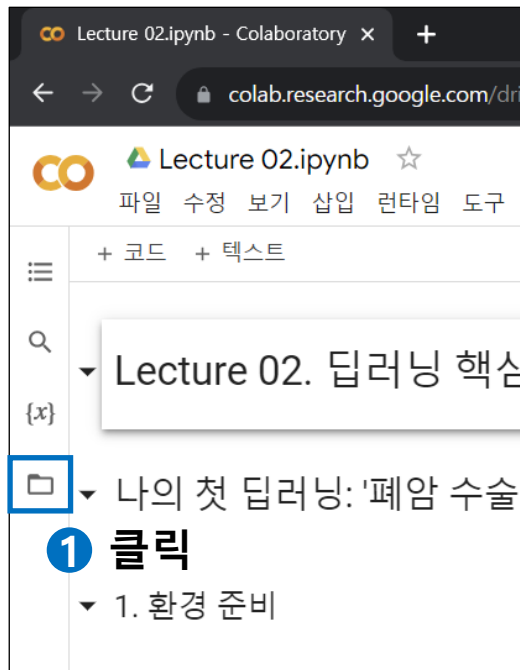
```
from (라이브러리명) import (새로운 이름)
```

- 예를 들어, ③ **import** numpy **as** np 명령은 아나콘다에 이미 포함되어 있는 넘파이(NumPy) 라이브러리를 np라는 짧은 이름으로 불러와 사용할 수 있게 해 줌

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ② 데이터 준비 (1/9)

- 이제 데이터를 불러와 구글 코랩에서 사용할 수 있도록 준비할 차례
- 데이터는 직접 업로드하는 방법과 깃허브에서 불러오는 방법이 있음
- 우리는 사전에 준비된 데이터를 다운로드하여 직접 업로드함
- ① 폴더 모양의 아이콘을 클릭한 후 ② data 폴더를 클릭하면 ③ 준비된 데이터를 확인할 수 있음



03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ② 데이터 준비 (2/9)

- data 폴더 안에 있는 데이터들은 ./data/데이터명 형식으로 불러올 수 있음
- 넘파이 라이브러리를 이용해 data 폴더에 있는 csv 파일을 불러오는 부분은 다음과 같음

```
dataset = np.loadtxt("./data/ThoracicSurgery3.csv", delimiter=',')
```

- Thoracic
a. 가슴(흉부)의
- Delimiter
n. 구획 문자

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ② 데이터 준비 (3/9)

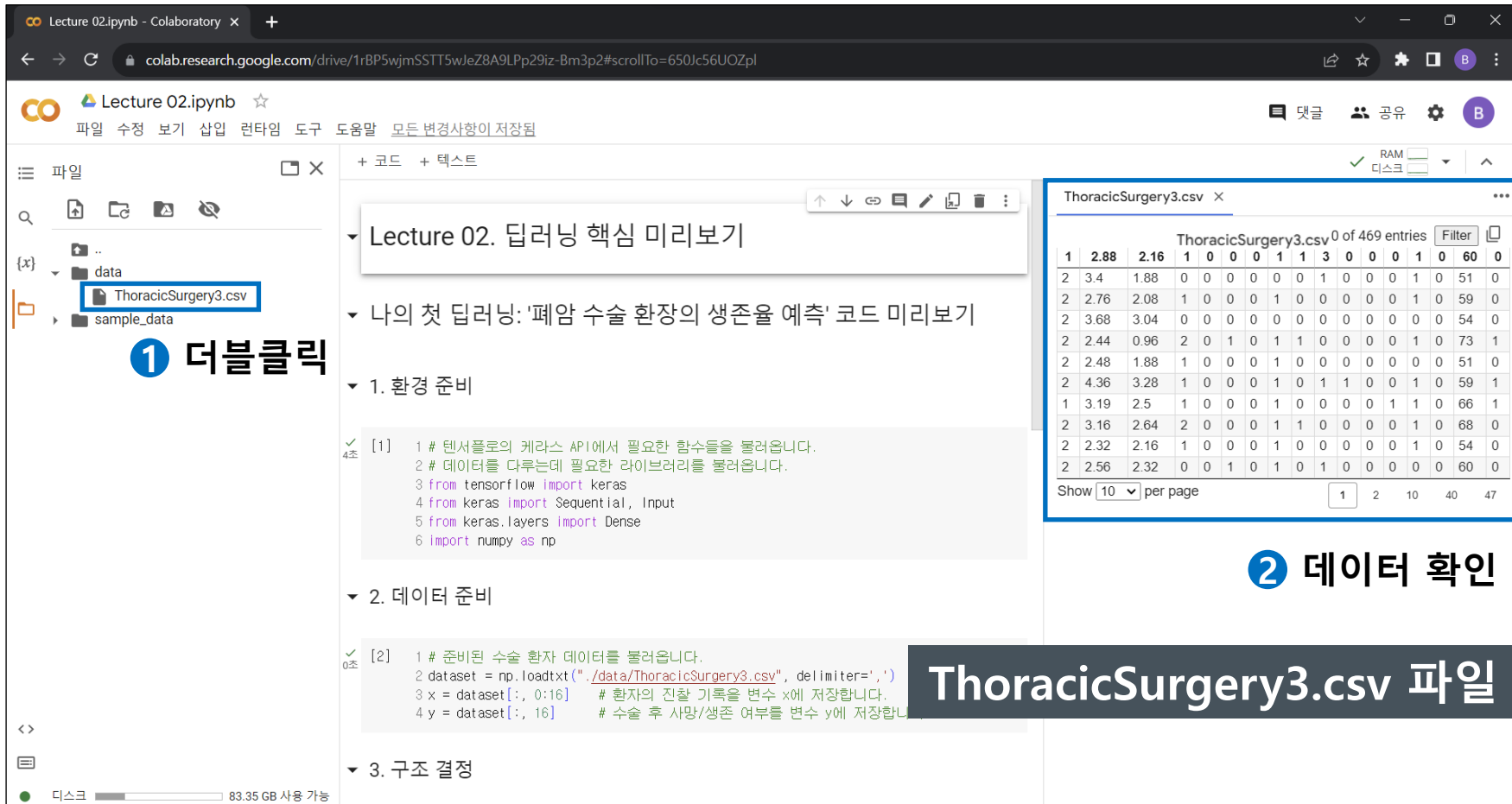
- 넘파이 라이브러리의 `loadtxt()` 함수를 사용해
'ThoracicSurgery3.csv'라는 외부 데이터셋을 불러왔음
- 머신러닝에서 알고리즘이나 좋은 컴퓨터 환경만큼 중요한 것이
바로 좋은 데이터를 준비하는 일
- 데이터를 면밀히 관찰하고 효율적으로 다루는 연습을 하는 것이 중요

불러온 ThoracicSurgery3.csv 파일에 관해
좀 더 살펴보겠습니다.

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ② 데이터 준비 (4/9)

- ① 먼저 아래 그림과 같이 data 폴더의 ThoracicSurgery3.csv 파일을 더블클릭
- ② 웹 브라우저 우측에 새로운 공간이 생기며 해당 데이터를 미리 볼 수 있음



1 더블클릭

2 데이터 확인

ThoracicSurgery3.csv 파일 확인

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ② 데이터 준비 (5/9)

폐암 수술 환자의 의료 기록과 1년 후 사망 여부 데이터

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	2.88	2.16	1	0	0	0	1	1	3	0	0	0	1	0	60	0
2	2	3.4	1.88	0	0	0	0	0	0	1	0	0	0	1	0	51	0
3	2	2.76	2.08	1	0	0	0	1	0	0	0	0	0	1	0	59	0
4	2	3.68	3.04	0	0	0	0	0	0	0	0	0	0	0	0	54	0
5	2	2.44	0.96	2	0	1	0	1	1	0	0	0	0	1	0	73	1
...
470	2	4.72	3.56	0	0	0	0	0	0	1	0	0	0	1	0	51	0

↑
샘플 수
(환자 수: 470명)

↑
속성
(의료 기록: 16가지)

↑
클래스
(사망: 0/생존: 1)

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ② 데이터 준비 (6/9)

- 가로줄 한 행이 한 사람의 환자로부터 기록된 정보를 의미
- 총 470행이므로 환자 470명에 대한 정보
- 한 행에는 17개의 숫자가 들어 있음
- 이는 환자마다 17개의 정보를 순서에 맞추어 정리했다는 의미
- 앞의 정보 16개는 종양의 유형, 폐활량, 호흡 곤란 여부, 고통 정도, 기침, 흡연, 천식 여부 등 16가지 환자 상태를 조사해서 기록해 놓은 것
- 마지막 17번째 정보는 수술 1년 후의 생존 결과
- 1은 수술 후 생존했음을, 0은 수술 후 사망했음을 의미

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ② 데이터 준비 (7/9)

- 이번 프로젝트의 목적은 1번째 항목부터 16번째 항목까지 이용해서 17번째 항목, 즉 수술 1년 후의 생존 또는 사망을 맞히는 것
- 1번째 항목부터 16번째 항목까지 속성(Attribute)이라 하고, 정답에 해당하는 17번째 항목을 클래스(Class 또는 Label)라고 함
- 클래스는 앞서 이야기한 '이름표'에 해당
- 딥러닝을 위해서는 속성과 클래스를 서로 다른 데이터셋으로 지정해 주어야 함

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ② 데이터 준비 (8/9)

- 먼저 속성으로 이루어진 데이터셋을 x라는 이름으로 만들어 줌

```
x = dataset[:, 0:16]
```

- ✓ 파이썬은 숫자를 1부터 세지 않고 0부터 셈
- ✓ 범위를 정할 경우 콜론(:) 앞의 숫자는 범위의 맨 처음을 의미하고, 콜론(:) 뒤의 숫자는 이 숫자가 가리키는 위치 '바로 앞'이 범위의 마지막이라는 의미
- ✓ 쉼표(,)를 기준으로 앞은 행(샘플), 뒤는 열(속성)의 범위가 입력
- ✓ 예를 들어,[:, 0:16]은 모든 행의 1번째 열부터 16번째 열까지 가져오라는 의미

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ② 데이터 준비 (9/9)

- 다음으로 17번째 줄에 위치한 클래스를 따로 모아 데이터셋 y로 지정

```
y = dataset[:, 16]
```


03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ③ 구조 결정 (1/6)

- 앞서 우리는 딥러닝을 실행시키기 위해 텐서플로를 불러왔음
- 텐서플로는 구글에서 만든 딥러닝 전용 라이브러리
- 텐서플로를 이용하면 여러 가지 알고리즘을 활용해 다양한 딥러닝 작업을 할 수 있지만, 사용법이 쉽지 않다는 단점이 있음



03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ③ 구조 결정 (2/6)

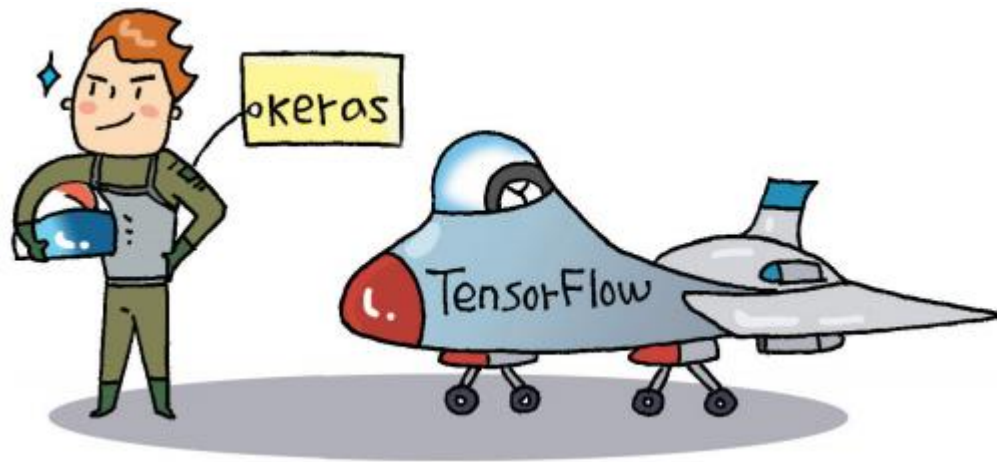
- 이를 해결해 주기 위해 개발된 것이 케라스(Keras)



03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ③ 구조 결정 (3/6)

- 텐서플로가 목적지까지 이동시켜 주는 비행기라면 케라스는 조종사에 해당
- 케라스를 활용하면 딥러닝의 거의 모든 작업을 쉽게 처리할 수 있음



텐서플로와 케라스의 관계

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ③ 구조 결정 (4/6)

- 불러온 예제에서 케라스를 어떻게 활용했는지 확인

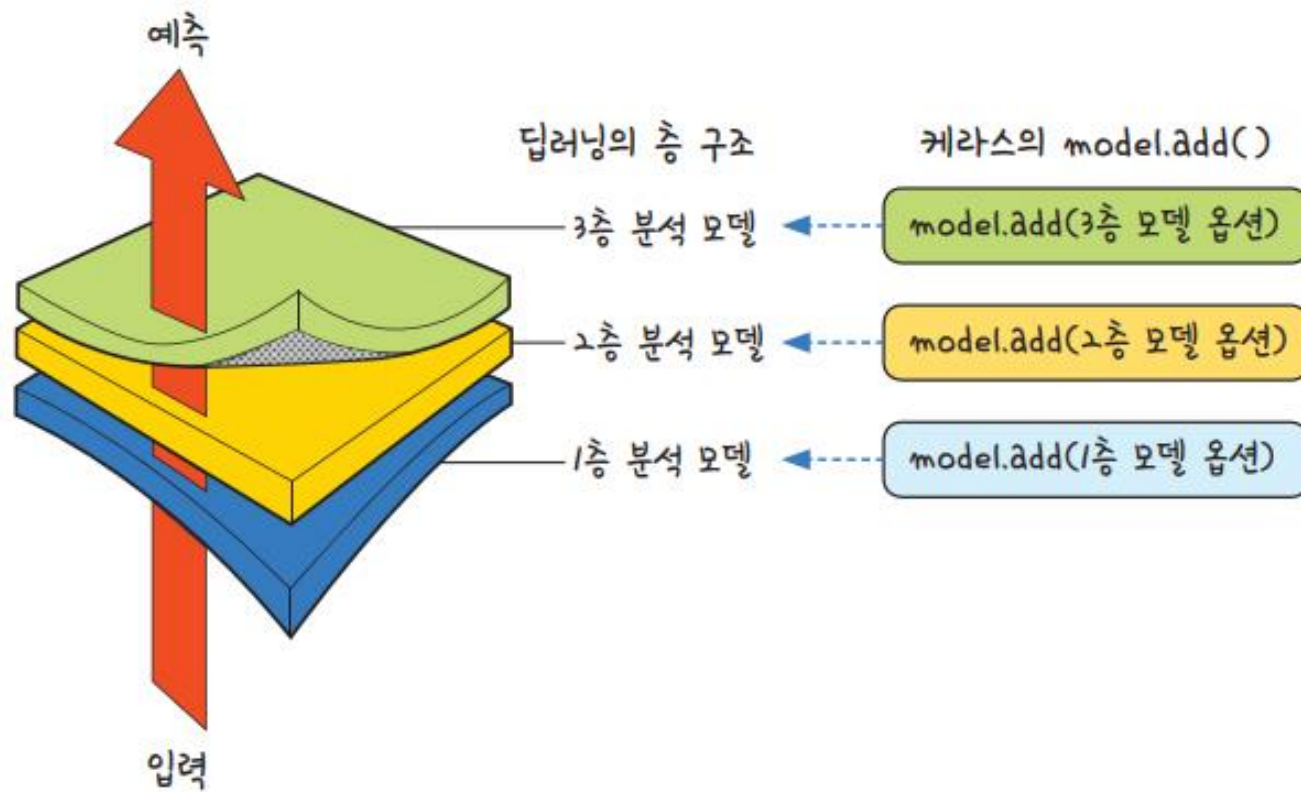
```
model = Sequential()  
model.add(Input(shape=(16,)))  
model.add(Dense(30, activation="relu"))  
model.add(Dense(1, activation="sigmoid"))
```

- ✓ 먼저 앞서 불러온 Sequential() 함수를 model로 선언
- ✓ 앞으로 상세히 다루겠지만, 딥러닝은 여러 층이 쌓여 있는 구조
- ✓ 준비된 데이터가 입력되는 입력층에 이어 첫 번째 작업을 진행하는 1층, 두 번째 작업을 하는 2층... 이런 식으로 출력 결과가 나오는 출력층까지, 여러 개의 층이 각자 자신이 맡은 일을 수행
- ✓ 케라스의 Sequential() 함수는 딥러닝의 한 층 한 층을 model.add()라는 함수를 사용해 추가시켜 줌
- ✓ 여기서는 입력층, 은닉층, 출력층으로 구성된 모델을 만들었음
- ✓ model.add() 함수를 한 줄 추가하는 것으로, 필요한 만큼 내부의 층을 만들 수 있음

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ③ 구조 결정 (5/6)

- 케라스 API의 layers 클래스에서 불러온 Dense() 함수가 model.add() 함수 안에서 호출되고 있음
- Dense는 '밀집한, 뾰뾰한'이란 뜻으로, 여기서는 각 층의 입력과 출력을 촘촘하게 모두 연결하라는 것



딥러닝의 층 구조와 케라스

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ③ 구조 결정 (6/6)

- 이제 두 가지를 더 알면 됨
 - 첫째, 좋은 딥러닝 모델을 만들려면 몇 개의 층으로 쌓아 올려야 하는가?
 - 둘째, Dense 함수 안에 있는 숫자와 설정의 의미는 무엇이며, 어떻게 정해야 하는가?
- ✓ 딥러닝을 설계한다는 것은 결국 몇 개의 층을 어떻게 쌓을지, 어떤 층을 사용할지, 내부의 변수들을 어떻게 정해야 하는지 등에 대해 고민하는 것
 - ✓ 대개 어떤 데이터를 가지고 무엇을 할 것인지에 따라 딥러닝의 설계가 결정
 - ✓ 각 설정과 변수의 의미를 알고 이것을 자유롭게 구성할 수 있는지가 딥러닝을 잘 다루는지 여부를 결정
 - ✓ 우리가 배울 내용도 결국 이것
 - ✓ Dense() 함수의 내부에 쓰인 각 설정의 의미들은 수업 진도가 나감에 따라 앞으로 하나씩 배우게 될 것

03. 딥러닝 개괄하기

❖ 소스 코드 리뷰: ④ 모델 실행

- 만들어 놓은 모델을 실행시키는 부분

```
model.compile(loss="binary_crossentropy",  
              optimizer="adam",  
              metrics=["accuracy"])  
history = model.fit(x, y, epochs=5, batch_size=16)
```

- ✓ model.compile() 함수는 앞서 만든 model의 설정을 그대로 실행하라는 의미
- ✓ 함수 내부에 loss, optimizer, metrics 등 키워드들이 들어 있음
- ✓ 이것은 앞 단계에서 만들어진 딥러닝 구조를 어떤 방식으로 구동시키고 어떻게 마무리할 것인지와 관련된 옵션들인데, 앞으로 자세히 배울 예정
- ✓ 딥러닝은 여러 층이 쌓여 만들어진다는 설명을 이미 한 바 있음
- ✓ 딥러닝의 기본 방식은 이 층들을 한 번만 통과하는 것이 아니라
- ✓ 위아래로 여러 차례 오가며 최적의 모델을 찾는 것
- ✓ 몇 번을 오갈 것인지, 그리고 한 번 오갈 때 몇 개의 데이터를 사용할 것인지
- ✓ 정하는 함수가 model.fit() 함수

04. 이제부터가 진짜 딥러닝?

- 01. 미지의 일을 예측하는 원리
- 02. 딥러닝 코드 실행해 보기
- 03. 딥러닝 개괄하기

04. 이제부터가 진짜 딥러닝?

❖ 지금까지 그리고 앞으로

- 지금까지 딥러닝을 위한 작업 환경을 만들고, 딥러닝 모델을 실행해 보면서 학습 목표를 파악했음
- 딥러닝을 위한 학습에는 단순한 파이썬 프로그래밍뿐만 아니라
선형 회귀, 로지스틱 회귀 등 기초 통계학 개념들도 필요함
- 이러한 설명에는 필연적으로 수학 개념이 따라오게 되어 있음

딥러닝을 위한 기초 수학 편을 통해
필요한 개념들을 정리하고 넘어갈 예정

- ❖ 01. 미지의 일을 예측하는 원리
- ❖ 02. 딥러닝 코드 실행해 보기
- ❖ 03. 딥러닝 개괄하기
- ❖ 04. 이제부터가 진짜 딥러닝?

THANK YOU!

Q & A

- Name: 권범
- Office: 동덕여자대학교 인문관 B821호
- Phone: 02-940-4752
- E-mail: bkwon@dongduk.ac.kr