Machine Learning & Deep Learning

Lecture 19

Keras를 이용한 딥러닝 코드 분석

(폐암 수술 환자의 생존율 예측)

김희숙 (H.S.Kim)

폐암 수술 환자의 생존율 예측 예제

Keras의 Sequential() 이용한 딥러닝

1. 딥러닝 모델 구조 설계 - add
   ① Dense : 은닉층, 출력층
   ② input_dim : 입력층
   ③ activation : 활성화 함수
2. 딥러닝 모델 컴파일 - compile
   ① loss : 오차 함수 정의
   ② optimizer : 기울기 하강법 정의
   ③ metrics : 정확도 정의
3. 딥러닝 모델 실행 - fit
   ① epochs : 전체 학습 횟수 정의
   ② batch_size : 한 번 학습시 샘플 개수 정의
4. 딥러닝 모델을 이용한 테스트 - predict
5. 딥러닝 모델 저장 - save
   ① h5 형식으로 딥러닝 결과 저장
6. 저장된 딥러닝 모델 읽기 - load_model
   ① 읽어온 h5 형식의 딥러닝 모델을 활용하여 미래 예측

Dr. Heesuk Kim

폴란드의 브로츠와프 의과대학에서 2013년 공개한 폐암 수술 환자의 수술 전 진단 데이터와 수술 후 생존 결과를 기록한 실제 의료 기록 데이터를 기반으로 딥러닝 코드를 분석해보자.

ThoracicSurgery.csv

01_ThoracicSurgery_deep_learning.ipynb

01_ThoracicSurgery_deep_learning.html

폐암 수술 환자의 생존율 예측 예제

Keras의 Sequential() 이용한 딥러닝

1. 딥러닝 모델 구조 설계 - add
   ① Dense : 은닉층, 출력층
   ② input_dim : 입력층
   ③ activation : 활성화 함수
2. 딥러닝 모델 컴파일 - compile
   ① loss : 오차 함수 정의
   ② optimizer : 기울기 하강법 정의
   ③ metrics : 정확도 정의
3. 딥러닝 모델 실행 - fit
   ① epochs : 전체 학습 횟수 정의
   ② batch_size : 한 번 학습시 샘플 개수 정의
4. 딥러닝 모델을 이용한 테스트 - predict
5. 딥러닝 모델 저장 - save
   ① h5 형식으로 딥러닝 결과 저장
6. 저장된 딥러닝 모델 읽기 - load_model
   ① 읽어온 h5 형식의 딥러닝 모델을 활용하여 미래 예측

Dr. Heesuk Kim

- 데이터셋 : **샘플**은 instance, example 으로 **속성**은 feature라고 부르기도 함

ThoracicSurgery.csv

```
# 준비된 수술 환자 데이터를 불러들입니다.
Data_set = np.loadtxt("./dataset/ThoracicSurgery.csv", delimiter=",")
```

```
Data_set
```

```
array([[293.  ,   1.  ,   3.8 , ...,   0.  ,  62.  ,   0.  ],
       [  1.  ,   2.  ,   2.88, ...,   0.  ,  60.  ,   0.  ],
       [  8.  ,   2.  ,   3.19, ...,   0.  ,  66.  ,   1.  ],
       ...,
       [406.  ,   6.  ,   5.36, ...,   0.  ,  62.  ,   0.  ],
       [ 25.  ,   8.  ,   4.32, ...,   0.  ,  58.  ,   1.  ],
       [447.  ,   8.  ,   5.2 , ...,   0.  ,  49.  ,   0.  ]])
```

속성 / 클래스

샘플	정보 1	정보 2	정보 3	…	정보 17	생존 여부
1번째 환자	293	1	3.8	…	62	0
2번째 환자	1	2	2.88	…	60	0
3번째 환자	8	3	3.19	…	66	1
…	…	…	…	…	…	…
470번째 환자	447	8	5.2	…	49	0

폐암 수술 환자의 생존율 예측 예제

Keras의 Sequential() 이용한 딥러닝

1. 딥러닝 모델 구조 설계 - add
   ① Dense : 은닉층, 출력층
   ② input_dim : 입력층
   ③ activation : 활성화 함수

2. 딥러닝 모델 컴파일 - compile
   ① loss : 오차 함수 정의
   ② optimizer : 기울기 하강법 정의
   ③ metrics : 정확도 정의

3. 딥러닝 모델 실행 - fit
   ① epochs : 전체 학습 횟수 정의
   ② batch_size : 한 번 학습시 샘플 개수 정의

4. 딥러닝 모델을 이용한 테스트 - predict

5. 딥러닝 모델 저장 - save
   ① h5 형식으로 딥러닝 결과 저장

6. 저장된 딥러닝 모델 읽기 - load_model
   ① 읽어온 h5 형식의 딥러닝 모델을 활용하여 미래 예측

Dr. Heesuk Kim

- 데이터셋 (100%) : 학습 데이터 셋 70%, 테스트 데이터 셋 30%로 분리

- **from sklearn.model_selection import train_test_split**

  사이킷런 모듈로 부터 **train_test_split**를 임포트한 상태에서 사용 가능함

```
# 환자의 기록과 수술 결과를 X와 Y로 구분하여 저장합니다.
X = Data_set[:,0:17]
Y = Data_set[:,17]
```

```
# 학습 데이터 70 %, 테스트 데이터셋 30% 로 설정하기
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=seed)
```

폐암 수술 환자의 생존율 예측 예제

Keras의 Sequential() 이용한 딥러닝

1. **딥러닝 모델 구조 설계 - add**
   - ① **Dense : 은닉층, 출력층**
   - ② **input_dim : 입력층**
   - ③ **activation : 활성화 함수**
2. 딥러닝 모델 컴파일 - compile
   - ① loss : 오차 함수 정의
   - ② optimizer : 기울기 하강법 정의
   - ③ metrics : 정확도 정의
3. 딥러닝 모델 실행 - fit
   - ① epochs : 전체 학습 횟수 정의
   - ② batch_size : 한 번 학습시 샘플 개수 정의
4. 딥러닝 모델을 이용한 테스트 - predict
5. 딥러닝 모델 저장 - save
   - ① h5 형식으로 딥러닝 결과 저장
6. 저장된 딥러닝 모델 읽기 - load_model
   - ① 읽어온 h5 형식의 딥러닝 모델을 활용하여 미래 예측


```
1  # 딥러닝 구조를 결정합니다(모델을 설정하고 실행하는 부분입니다).
2  model = Sequential()
3  model.add(Dense(30, input_dim=17, activation='relu'))
4  model.add(Dense(1, activation='sigmoid'))
```

- 딥러닝이란 퍼셉트론 위에 **숨겨진 퍼셉트론** 층을 차곡차곡 추가하는 형태
- 이 층들이 케라스에서는 **Sequential()** 함수를 통해 쉽게 구현됨
- **Sequential()** 함수를 **model**로 선언해 놓고 **model.add()**라는 라인을 추가하면 새로운 층이 만들어짐

  → 코드에는 **model.add()로 시작되는 라인이 2개**가 있으므로 2개의 층을 가진 모델을 만든다는 것
- 맨 마지막 층은 결과를 출력하는 '출력층'이 됨
- 나머지는 모두 '은닉층'의 역할을 함
- 각각의 층은 **Dense**라는 함수를 통해 구체적으로 그 구조가 결정됨

폐암 수술 환자의 생존율 예측 예제

Keras의 Sequential() 이용한 딥러닝

1. **딥러닝 모델 구조 설계 - add**
   - ① **Dense : 은닉층, 출력층**
   - ② **input_dim : 입력층**
   - ③ **activation : 활성화 함수**
2. 딥러닝 모델 컴파일 - compile
   - ① loss : 오차 함수 정의
   - ② optimizer : 기울기 하강법 정의
   - ③ metrics : 정확도 정의
3. 딥러닝 모델 실행 - fit
   - ① epochs : 전체 학습 횟수 정의
   - ② batch_size : 한 번 학습시 샘플 개수 정의
4. 딥러닝 모델을 이용한 테스트 - predict
5. 딥러닝 모델 저장 - save
   - ① h5 형식으로 딥러닝 결과 저장
6. 저장된 딥러닝 모델 읽기 - load_model
   - ① 읽어온 h5 형식의 딥러닝 모델을 활용하여 미래 예측

Dr. Heesuk Kim

```
1  # 딥러닝 구조를 결정합니다(모델을 설정하고 실행하는 부분입니다).
2  model = Sequential()
3  model.add(Dense(30, input_dim=17, activation='relu'))
4  model.add(Dense(1, activation='sigmoid'))
```

- **model.add(Dense(30, input_dim=17, activation='relu'))**

  → 이 층에 몇 개의 노드를 만들 것인지를 결정

  - ✓ **첫 번째 층의 Dense**는 **입력층**과 **첫 번째 은닉층**을 구성함
  - ✓ **30 이라고 되어 있는 것은 이 층에서 출력되는 은닉층 노드를 30개 만들겠다는 것**
  - ✓ **input_dim** 변수는 **입력 데이터**로부터 **몇 개의 입력되는지를** 정하는 것
    - ➢ 폐암 수술 환자의 생존 여부 데이터에는 **17개의 입력 값**이 있음
  - ✓ **입력 데이터에서 17개의 값을** 받아 **은닉층의 30개 노드로** 보낸다는 뜻
  - ✓ **activation** 부분에 원하는 활성화 함수를 적어 주면 됨 → **렐루**를 사용함
- **model.add(Dense(1, activation='sigmoid'))**
  - ✓ **두 번째 Dense**는 **출력층**을 의미하고 **출력 노드는 1개**
  - ✓ **activation** 부분에 원하는 활성화 함수를 적어 주면 됨 → **시그모이드(sigmoid)** 를 사용함

폐암 수술 환자의 생존율 예측 예제

Keras의 Sequential() 이용한 딥러닝

1. 딥러닝 모델 구조 설계 - add
   ① Dense : 은닉층, 출력층
   ② input_dim : 입력층
   ③ activation : 활성화 함수

2. **딥러닝 모델 컴파일 - compile**
   ① loss : 오차 함수 정의
   ② optimizer : 기울기 하강법 정의
   ③ metrics : 정확도 정의

3. 딥러닝 모델 실행 - fit
   ① epochs : 전체 학습 횟수 정의
   ② batch_size : 한 번 학습시 샘플 개수 정의

4. 딥러닝 모델을 이용한 테스트 - predict

5. 딥러닝 모델 저장 - save
   ① h5 형식으로 딥러닝 결과 저장

6. 저장된 딥러닝 모델 읽기 - load_model
   ① 읽어온 h5 형식의 딥러닝 모델을 활용하여 미래 예측

Dr. Heesuk Kim

```
1  # 딥러닝을 실행합니다.
2  model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])
3  #model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
4  model.fit(X_train, Y_train, epochs=100, batch_size=10) # 학습 데이터셋 329 개로 학습
5
```

- **model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])**
- **model.compile** 부분은 앞서 지정한 모델이 효과적으로 구현될 수 있게 여러 가지 환경을 설정해 주면서 컴파일하는 부분
  - ✓ **loss : 오차(loss) 함수** 부분 결정 → **평균 제곱 오차 함수(mean_squared_error, MSE)**를 사용
  - ✓ **optimizer : 기울기 하강법** 부분 결정 → 최적화를 위해 **아담(adam)**을 사용
  - ✓ **metrics() :** 함수는 모델이 컴파일될 때 모델 수행 결과를 출력하는 부분
    - ➢ 정확도를 측정하기 위해 사용되는 **테스트 샘플을 학습 과정에서 제외**시킴으로써 **과적합 문제를 방지**하는 기능을 담고 있음

폐암 수술 환자의 생존율 예측 예제

Keras의 Sequential() 이용한 딥러닝

1. 딥러닝 모델 구조 설계 - add
   ① Dense : 은닉층, 출력층
   ② input_dim : 입력층
   ③ activation : 활성화 함수

2. 딥러닝 모델 컴파일 - compile
① loss : 오차 함수 정의
② optimizer : 기울기 하강법 정의
③ metrics : 정확도 정의

3. 딥러닝 모델 실행 - fit
   ① epochs : 전체 학습 횟수 정의
   ② batch_size : 한 번 학습시 샘플 개수 정의

4. 딥러닝 모델을 이용한 테스트 - predict

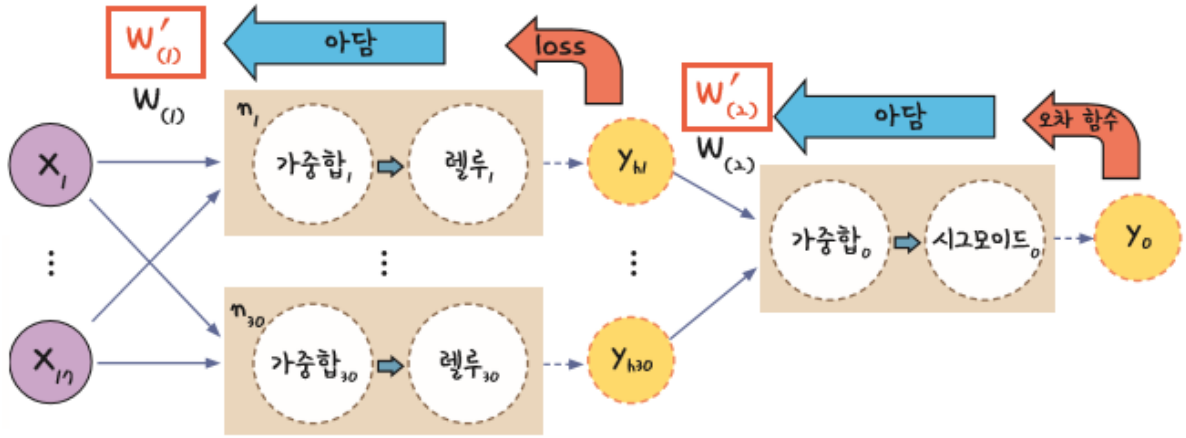
5. 딥러닝 모델 저장 - save
   ① h5 형식으로 딥러닝 결과 저장

6. 저장된 딥러닝 모델 읽기 - load_model
   ① 읽어온 h5 형식의 딥러닝 모델을 활용하여 미래 예측


```
# 딥러닝을 실행합니다.
model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])
#model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X_train, Y_train, epochs=100, batch_size=10) # 학습 데이터셋 329 개로 학습

```

- **model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])**



폐암 수술 환자의 생존율 예측 예제

Keras의 Sequential() 이용한 딥러닝

1. 딥러닝 모델 구조 설계 - add
   ① Dense : 은닉층, 출력층
   ② input_dim : 입력층
   ③ activation : 활성화 함수

2. 딥러닝 모델 컴파일 - compile
① loss : 오차 함수 정의
② optimizer : 기울기 하강법 정의
③ metrics : 정확도 정의

3. 딥러닝 모델 실행 - fit
   ① epochs : 전체 학습 횟수 정의
   ② batch_size : 한 번 학습시 샘플 개수 정의

4. 딥러닝 모델을 이용한 테스트 - predict

5. 딥러닝 모델 저장 - save
   ① h5 형식으로 딥러닝 결과 저장

6. 저장된 딥러닝 모델 읽기 - load_model
   ① 읽어온 h5 형식의 딥러닝 모델을 활용하여 미래 예측

Dr. Heesuk Kim

```
1  # 딥러닝을 실행합니다.
2  #model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])
3  model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
4  model.fit(X_train, Y_train, epochs=100, batch_size=10) # 학습 데이터셋 329 개로 학습
5
```

- **model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])**
  - ✓ **loss : 오차(loss) 함수** 부분

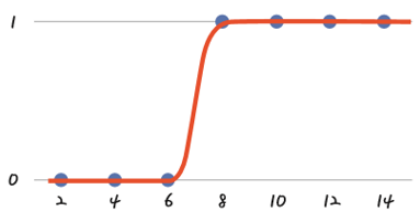
    오차 함수에는 **평균 제곱 오차 함수(mean_squared_error, MSE)** 계열의 함수 외에

    시그모이드를 사용하여 0 또는 1로 분류할 때 사용하는 **'binary_crossentropy'** 와 **같은**

    교차 엔트로피(Cross Entropy) 계열의 함수가 있다.

    - ➢ **교차 엔트로피(Cross Entropy)**는 출력 값에 **로그**를 취해서,

      오차가 커지면 **수렴 속도가 빨라지고, 오차가 작아지면 속도가 감소**하도록 만든 함수



시그모이드 함수 그래프

loss

$$cost\,(H(x), y) = -\{\underbrace{y \log h}_{A} + \underbrace{(1-y)\log(1-h)}_{B}\}$$

폐암 수술 환자의 생존율 예측 예제

- Keras에서 사용되는 대표적인 **loss**(오차) 함수

실제 값을 yt, 예측 값을 yo라고 가정할 때

평균 제곱 계열	mean_squared_error	평균 제곱 오차 계산: mean(square(yt - yo))
	mean_absolute_error	평균 절대 오차(실제 값과 예측 값 차이의 절댓값 평균) 계산: mean(abs(yt - yo))
	mean_absolute_percentage_error	평균 절대 백분율 오차(절댓값 오차를 절댓값으로 나눈 후 평균) 계산: mean(abs(yt - yo)/abs(yt)) (단, 분모 ≠ 0)
	mean_squared_logarithmic_error	평균 제곱 로그 오차(실제 값과 예측 값에 로그를 적용한 값의 차이를 제곱한 값의 평균) 계산: mean(square((log(yo) + 1) - (log(yt) + 1)))
교차 엔트로피 계열	categorical_crossentropy	범주형 교차 엔트로피(일반적인 분류)
	binary_crossentropy	이항 교차 엔트로피(두 개의 클래스 중에서 예측할 때)

폐암 수술 환자의 생존율 예측 예제

Keras의 Sequential() 이용한 딥러닝

1. 딥러닝 모델 구조 설계 - add
   ① Dense : 은닉층, 출력층
   ② input_dim : 입력층
   ③ activation : 활성화 함수

2. 딥러닝 모델 컴파일 - compile
   ① loss : 오차 함수 정의
   ② optimizer : 기울기 하강법 정의
   ③ metrics : 정확도 정의

3. 딥러닝 모델 실행 - fit
① epochs : 전체 학습 횟수 정의
② batch_size : 한 번 학습시 샘플 개수 정의

4. 딥러닝 모델을 이용한 테스트 - predict

5. 딥러닝 모델 저장 - save
   ① h5 형식으로 딥러닝 결과 저장

6. 저장된 딥러닝 모델 읽기 - load_model
   ① 읽어온 h5 형식의 딥러닝 모델을 활용하여 미래 예측

Dr. Heesuk Kim

```
# 딥러닝을 실행합니다.
#model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X_train, Y_train, epochs=100, batch_size=10) # 학습 데이터셋 329 개로 학습

```

- **model.fit(X, Y, epochs=100, batch_size=10)**
  - ✓ **epochs :** 에포크
    - ➢ 학습 프로세스가 모든 샘플에 대해 한 번 실행되는 것
    - ➢ 코드에서 **epochs=100** 으로 지정한 것은 각 데이터 학습 데이터 329개가 100번 학습될 때까지 실행을 반복하라는 뜻
  - **batch_size**
    - ➢ 샘플을 한 번에 몇 개씩 처리할지를 정하는 부분으로 **batch_size=10** 은 전체 329개의 학습 데이터를 10개씩 끊어서 학습시킨다는 뜻
    - ➢ **batch_size**가 너무 크면 학습 속도가 느려지고, 너무 작으면 각 실행 값의 편차가 생겨서 전체 결과값이 불안정해질 수 있음
    - ➢ 자신의 컴퓨터 **메모리**가 감당할 만큼의 **batch_size**를 찾아 설정해 주는 것이 좋음

폐암 수술 환자의 생존율 예측 예제


```
1  # 테스트 데이터셋으로 정확도를 계산합니다.
2
3  print("₩n Test Accuracy : %.4f" % (model.evaluate(X_test, Y_test)[1]))
```

- **model.evaluate(X_test, Y_test)[1]**
  - ✓ 테스트 데이터셋 141개를 이용하여 딥러닝 모델의 정확도를 계산

폐암 수술 환자의 생존율 예측 예제

▪ 테스트 데이터 예측 : model.predict()

✓ **속성**에 해당하는 부분만 입력하면 딥러닝 프로그램이 **클래스**를 예측함

```
1  import numpy as np
2  kim = np.array([[293,1,3.8,2.8,0,0,0,0,0,0,12,0,0,0,1,0,62]])
3  print(model.predict(kim))
4
5  test_kim  = model.predict(kim)*100
6  print("백분율로 표시 : ",test_kim)
7  print("Kim 폐암 수술 후 생존율 예측 : %.2f" %test_kim, "%")
```

```
[[0.03467155]]
백분율로 표시 :  [[3.4671545]]
Kim 폐암 수술 후 생존율 예측 : 3.47 %
```

```
1  park = np.array([[197,3,2.84,2.24,1,1,1,0,0,0,12,0,0,0,1,0,68]])
2  print(model.predict(park))
3
4
5  park_predict = model.predict(park)*100
6  print("백분율로 표시 : ",park_predict)
7  print("park 폐암 수술 후 생존율 예측 : %.2f" %park_predict, "%")
```

```
[[0.10510644]]
백분율로 표시 :  [[10.510644]]
park 폐암 수술 후 생존율 예측 : 10.51 %
```

폐암 수술 환자의 생존율 예측 예제

Dr. Heesuk Kim

- **딥러닝 모델 저장 : model.save('ooooo.h5')**
  - ✓ 학습 데이터 셋으로 학습한 결과를 h5 형식으로 저장함

```
1  model.save('01_ThoracicSurgery_deep_learning_result_save.h5')
```

폐암 수술 환자의 생존율 예측 예제

Keras의 Sequential() 이용한 딥러닝

1. 딥러닝 모델 구조 설계 - add
   ① Dense : 은닉층, 출력층
   ② input_dim : 입력층
   ③ activation : 활성화 함수
2. 딥러닝 모델 컴파일 - compile
   ① loss : 오차 함수 정의
   ② optimizer : 기울기 하강법 정의
   ③ metrics : 정확도 정의
3. 딥러닝 모델 실행 - fit
   ① epochs : 전체 학습 횟수 정의
   ② batch_size : 한 번 학습시 샘플 개수 정의
4. 딥러닝 모델을 이용한 테스트 - predict
5. 딥러닝 모델 저장 - save
   ① h5 형식으로 딥러닝 결과 저장
6. **저장된 딥러닝 모델 읽기 - load_model**
   ① **읽어온 h5 형식의 딥러닝 모델을 활용하여 미래 예측**

▪ 저장된 딥러닝 모델 읽어오기 : load.model('ooooo.h5')

✓ 저장되어 있는 **h5** 형식의 딥러닝 모델을 읽어와서 다시 모델을 셋팅한 후,

속성에 해당하는 입력 데이터를 기반으로 **model.predict()을 사용하여 클래스**를 예측

```
from tensorflow import keras
from tensorflow.keras import layers
from keras.models import load_model
model = keras.models.load_model('01_ThoracicSurgery_deep_learning_result_save.h5')
```

```
import numpy as np
kim = np.array([[293,1,3.8,2.8,0,0,0,0,0,0,12,0,0,0,1,0,62]])
print(model.predict(kim))

test_kim  = model.predict(kim)*100
print("백분율로 표시 : ",test_kim)
print("Kim 폐암 수술 후 생존율 예측 : %.2f" %test_kim, "%")
```

```
[[0.03467155]]
백분율로 표시 :  [[3.4671545]]
Kim 폐암 수술 후 생존율 예측 : 3.47 %
```

저장된 모델을 기반으로 앞에서 사용한 테스트 값으로 예측을 실행하였으므로 앞에서 예측한 것과 결과는 동일함

다른 새로운 에디터에서 실습하기!!

```
park = np.array([[197,3,2.84,2.24,1,1,1,0,0,0,12,0,0,0,1,0,68]])
print(model.predict(park))


park_predict = model.predict(park)*100
print("백분율로 표시 : ",park_predict)
print("park 폐암 수술 후 생존율 예측 : %.2f" %park_predict, "%")
```

```
[[0.10510644]]
백분율로 표시 :  [[10.510644]]
park 폐암 수술 후 생존율 예측 : 10.51 %
```

저장된 모델을 기반으로 앞에서 사용한 테스트 값으로 예측을 실행하였으므로 앞에서 예측한 것과 결과는 동일함