

2.2 기본예제들

(책 3. 신경망 시작하기 참고)

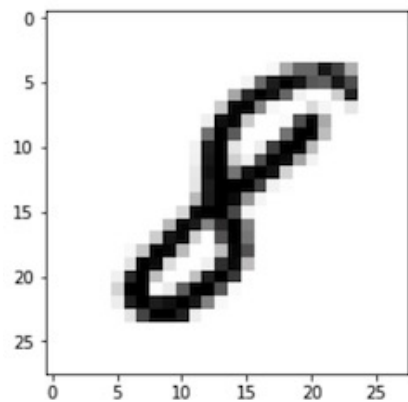
개요

- 간단한 딥러닝 프로그램 예제를 돌려보고, 이에 익숙해진다
- 이진 분류(Binary Classification) : 영화 리뷰 (좋다/나쁘다)
- 다중 분류(Multiclass classification) : 뉴스 기사 분류 (정치, 사회, 스포츠, 문화, 영화, ...)
- 회귀 분석 (Regression) : 미래 집값 예측

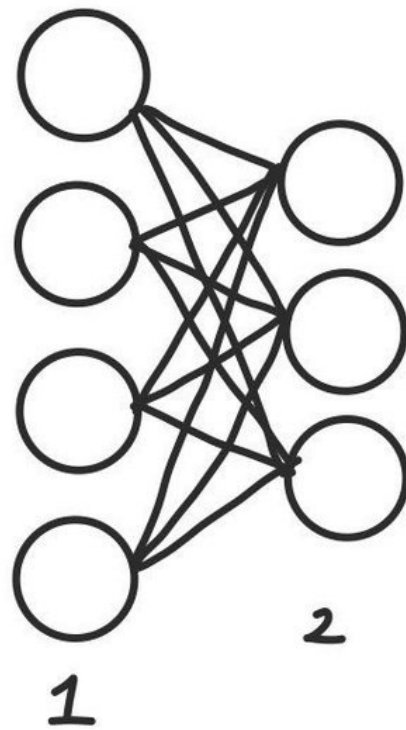
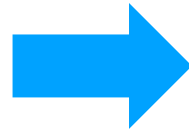
프로세스

- 데이터셋 (어떤 데이터셋인가?, 어떻게 구성되어 있나?)
- 데이터 전처리
- 모델 수립
- 학습/검증
- 시험/평가/가시화

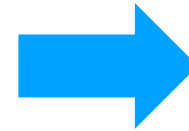
복습: MNIST 데이터셋 학습



데이터셋
+
전처리



모델 수립



Value

이게 얼마가 될거 같니?

output을
그냥 받는다.

MSE: Mean Squared Error

O/X

기냐? 아니냐?

output에
sigmoid를 먹인다.

Binary CrossEntropy

Category

종류중에 요건 뭐냐?

output에
softmax를 먹인다.

Categorical CrossEntropy

Loss 함수 결정

```
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
```

```
[ ] train_images.shape
```

👤 (60000, 28, 28)

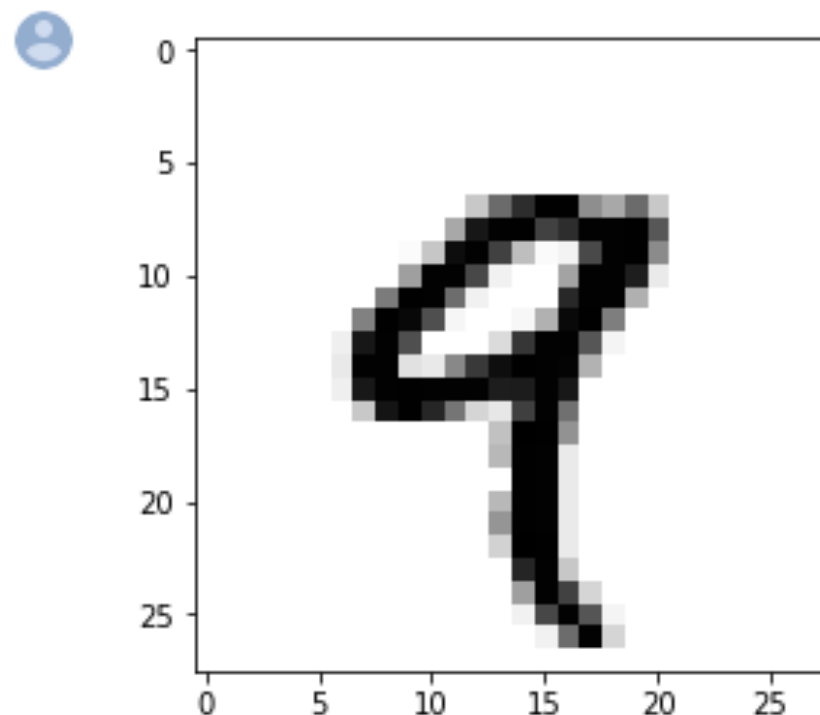
▶ train_labels

👤 array([5, 0, 4, ..., 5, 6, 8], dtype=uint8)

```
[ ] import matplotlib.pyplot as plt
```

```
[ ] digit = train_images[4]
```

```
plt.imshow(digit, cmap=plt.cm.binary)
plt.show()
```



```
network = models.Sequential()  
network.add(layers.Dense(512, activation='relu', input_shape=(28 * 28,)))  
network.add(layers.Dense(10, activation='softmax'))
```

```
[ ] network.compile(optimizer='rmsprop',  
                    loss='categorical_crossentropy',  
                    metrics=['accuracy'])
```

```
[ ] train_images = train_images.reshape((60000, 28 * 28))
    train_images = train_images.astype('float32') / 255

    test_images = test_images.reshape((10000, 28 * 28))
    test_images = test_images.astype('float32') / 255
```

```
train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)
```

이제 신경망을 훈련시킬 준비가 되었습니다. 케라스에서는 `fit` 메서드를 호출하여 훈련 데이터에 모델을 학습시킵니다:

```
[ ] network.fit(train_images, train_labels, epochs=5, batch_size=128)
```



```
Epoch 1/5
60000/60000 [=====] - 1s 22us/step - loss: 0.2571 - acc: 0.9257
Epoch 2/5
60000/60000 [=====] - 1s 12us/step - loss: 0.1027 - acc: 0.9695
Epoch 3/5
60000/60000 [=====] - 1s 12us/step - loss: 0.0686 - acc: 0.9797
Epoch 4/5
60000/60000 [=====] - 1s 12us/step - loss: 0.0494 - acc: 0.9856
Epoch 5/5
60000/60000 [=====] - 1s 12us/step - loss: 0.0368 - acc: 0.9894
<keras.callbacks.History at 0x7f3aa51c3908>
```

훈련하는 동안 두 개의 정보가 출력됩니다. 훈련 데이터에 대한 네트워크의 손실과 정확도입니다.

훈련 데이터에 대해 0.989(98.9%)의 정확도를 금방 달성합니다. 이제 테스트 세트에서도 모델이 잘 작동하는지 확인해 보겠습니다:

```
[ ] test_loss, test_acc = network.evaluate(test_images, test_labels)
```

```
10000/10000 [=====] - 0s 16us/step
```

```
[ ] print('test_acc:', test_acc)
```

```
test_acc: 0.9789
```


tf.keras.datasets

Modules

`boston_housing` module: Boston housing price regression dataset.

`cifar10` module: CIFAR10 small images classification dataset.

`cifar100` module: CIFAR100 small images classification dataset.

`fashion_mnist` module: Fashion-MNIST dataset.

`imdb` module: IMDB sentiment classification dataset.

`mnist` module: MNIST handwritten digits dataset.

`reuters` module: Reuters topic classification dataset.

1. 영화 리뷰 분류 : 이진 분류

- IMDB (Internet Movie Database)는 영화 소개 및 사용자 리뷰를 볼 수 있는 미국의 유명 사이트
- IMDB Dataset : 총 50,000개의 영화리뷰를 포함하고 있으며, 해당 영화 리뷰에 대한 긍정/부정(binary sentiment)을 가지고 있는 데이터 셋으로 스탠포드 대학에서 수집/전처리

FULL CAST AND CREW

TRIVIA

USER REVIEWS

IMDbPro

MORE

SHARE



Avengers: Endgame (2019)

★ 8.6^{/10}
561,070★ Rate
This

12 | 3h 1min | Action, Adventure, Sci-Fi | 24 April 2019 (South Korea)



1:06 | Trailer

115 VIDEOS | 879 IMAGES

After the devastating events of [Avengers: Infinity War](#) (2018), the universe is in ruins. With the help of remaining allies, the Avengers assemble once more in order to reverse Thanos' actions and restore balance to the universe.

Directors: [Anthony Russo](#), [Joe Russo](#)

Writers: [Christopher Markus](#) (screenplay by), [Stephen McFeely](#) (screenplay by) | [4 more credits](#) »

Stars: [Robert Downey Jr.](#), [Chris Evans](#), [Mark Ruffalo](#) | [See full cast & crew](#) »

[+ Add to Watchlist](#)

78

Metascore
From [metacritic.com](#)

Reviews
8,116 user | 519 critic



Popularity
17 (↓ 6)

★ 10/10

The End of an Era!

[ahmetkozan](#) 25 April 2019

After Avengers Infinity War, we waited for the Avengers Endgame. We wondered how the story would go on, how our heroes would turn back, what would be the end of Thanos. Many theories related to this have been put forward. Avengers Endgame was undoubtedly the most anticipated film of recent years. Normally, the higher the expectation, the higher the probability of disappointment. But this is not the case for Endgame. Whatever you're expecting, you find much more in the film. This means that the biggest concern about the film has disappeared.

On the other hand, another comparison comes up. Is Endgame more successful than Infinity War? We can comfortably say it Avengers Infinity War is just the beginning of the story. Endgame was the finale of the story. So we shouldn't think of these two films as

1,831 out of 3,158 found this helpful. Was this review helpful? [Sign in](#) to vote.

[Permalink](#)

★ 6/10

Amazing, but the more I dwell on it the worse it becomes

[theroo200182](#) 26 April 2019

Warning: Spoilers

2,620 out of 4,553 found this helpful. Was this review helpful? [Sign in](#) to vote.

[Permalink](#)

★ 3/10

Disappointing experience

[ircolist](#) 16 August 2019

If you ignored the PC-stuff in the last movie, it was still a pretty enjoyable movie. But this? Nothing was coherent, it was too long, and it it wasn't until the very end that it delieved any decent action.

59 out of 89 found this helpful. Was this review helpful? [Sign in](#) to vote.

[Permalink](#)

IMDB Dataset.csv (63.14 MB)

	A review	A sentiment
	49582 unique values	positive 50% negative 50%
1	One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. They are right, as this is exactly what happened with me. The first thing that struck me abo...	positive
2	A wonderful little production. The filming technique is very unassuming- very old-time-BBC fashion and gives a comforting, and sometimes discomforting, sense of realism to the entire piece...	positive

가 가

3	I thought this was a wonderful way to spend time on a too hot summer weekend, sitting in the air conditioned theater and watching a light-hearted comedy. The plot is simplistic, but the dialogue is wi...	positive
4	Basically there's a family where a little boy (Jake)	negative

0 1 1

데이터 셋

```
from keras.datasets import imdb  
  
(train_data, train_labels), (test_data, test_labels) = imdb.load_data(num_words=10000)
```

- Keras에서 기본 데이터 셋으로 제공
- num_words는 영화리뷰에 나타난 단어 중 가장 자주 나타나는 단어 10000개만 사용하겠다는 의미
가 one-hot-encoding
- 핵심: 어떻게 각 리뷰 데이터를 고차원의 점 하나로 표현

전처리

- 각 리뷰 데이터는 10,000개의 단어만을 사용
- 하지만, 각 리뷰의 길이는 서로 다르다
- 모델에 입력하는 input의 길이는 일정하다. 해결책은?

One-Hot Encoding



color	color_red	color_blue	color_green
red	1	0	0
green	0	0	1
blue	0	1	0
red	1	0	0

0,1

One-Hot Encoding

```
import numpy as np

def vectorize_sequences(sequences, dimension=10000):
    # 크기가 (len(sequences), dimension)이고 모든 원소가 0인 행렬을 만듭니다
    results = np.zeros((len(sequences), dimension))
    for i, sequence in enumerate(sequences):
        results[i, sequence] = 1. # results[i]에서 특정 인덱스의 위치를 1로 만듭니다
    return results

# 훈련 데이터를 벡터로 변환합니다
x_train = vectorize_sequences(train_data)
# 테스트 데이터를 벡터로 변환합니다
x_test = vectorize_sequences(test_data)
```

- 차원의 개수만큼 공간을 할당하고 미리 0으로 초기화
- 리뷰의 각 단어에 해당하는 위치만 1로 세팅한다.

전처리 : One-Hot Encoding 후

```
# 훈련 데이터를 벡터로 변환합니다
x_train = vectorize_sequences(train_data)
# 테스트 데이터를 벡터로 변환합니다
x_test = vectorize_sequences(test_data)
```

이제 샘플은 다음과 같이 나타납니다:

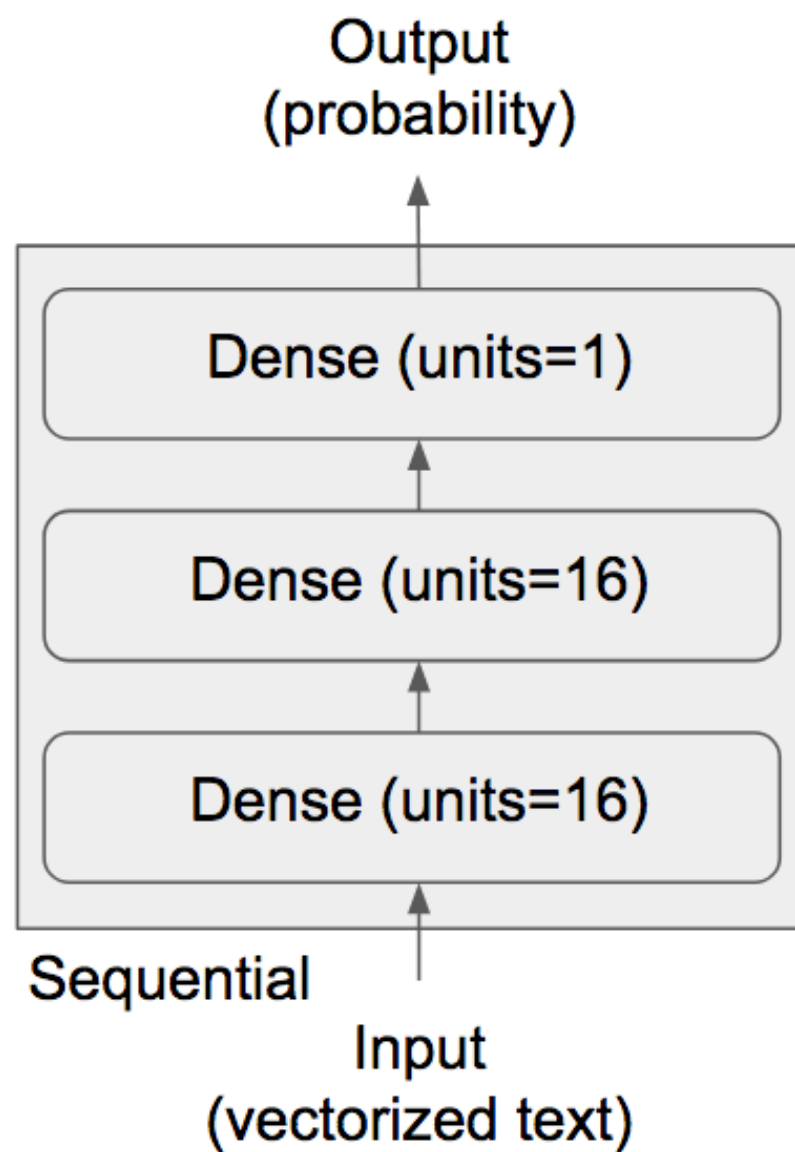
```
x_train[0]
```

```
array([0., 1., 1., ..., 0., 0., 0.])
```

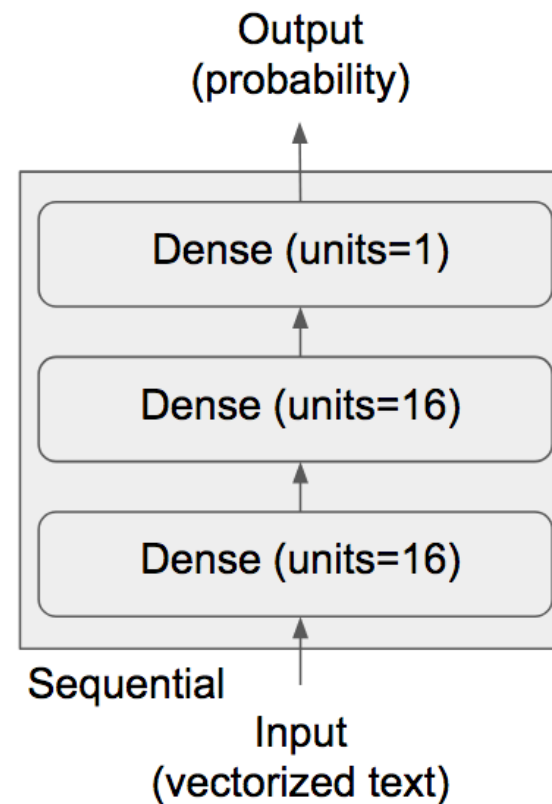
신경망 모델 만들기

- 모델을 코딩해 보자
- `m = models.Sequential()`
- `m.add(layers.Dense(...))`
- ...
- 입력 차원은 10000 차원의 데이터 (One-hot)
- 출력 차원은 1차원 (Positive일 확률)

신경망 모델 만들기



신경망 모델 만들기



```
from keras import models
from keras import layers

model = models.Sequential()
model.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))
```

0 1

sigmoid

최적화

- Optimizer는?
- Loss 함수는?
- Metrics는?

최적화

- Optimizer는?
- Loss 함수는? 가 가 O,X binary_crossentropy
- Metrics는?

```
model.compile(optimizer='rmsprop',  
              loss='binary_crossentropy',  
              metrics=['accuracy'])
```

학습

```
x_val = x_train[:10000]
partial_x_train = x_train[10000:]

y_val = y_train[:10000]
partial_y_train = y_train[10000:]
```

```
history = model.fit(partial_x_train,
                    partial_y_train,
                    epochs=20,
                    batch_size=512,
                    validation_data=(x_val, y_val))
```

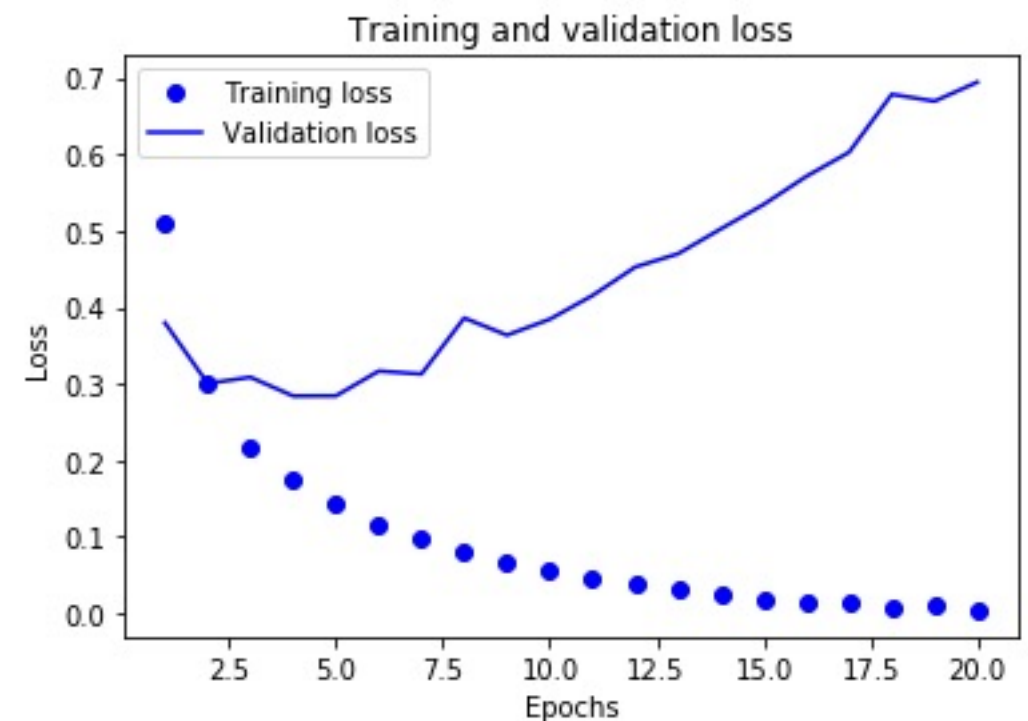

학습 결과 가시화

```
history_dict = history.history  
history_dict.keys()
```

```
dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
```

```
import matplotlib.pyplot as plt
```

```
acc = history.history['acc']  
val_acc = history.history['val_acc']  
loss = history.history['loss']  
val_loss = history.history['val_loss']  
  
epochs = range(1, len(acc) + 1)  
  
# 'bo'는 파란색 점을 의미합니다  
plt.plot(epochs, loss, 'bo', label='Training loss')  
# 'b'는 파란색 실선을 의미합니다  
plt.plot(epochs, val_loss, 'b', label='Validation loss')  
plt.title('Training and validation loss')  
plt.xlabel('Epochs')  
plt.ylabel('Loss')  
plt.legend()  
  
plt.show()
```

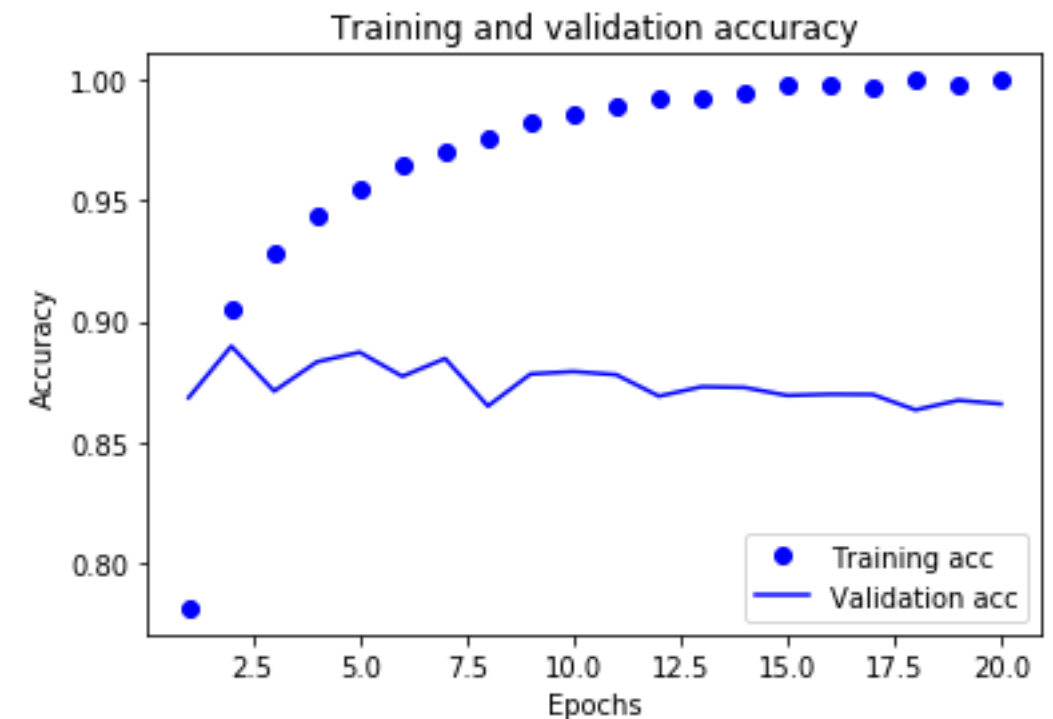


학습 결과 가시화

```
plt.clf() # 그래프를 초기화합니다
acc = history_dict['acc']
val_acc = history_dict['val_acc']

plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()

plt.show()
```



(개별실습) 데이터 셋

```
>>> train_data[0]
[1, 14, 22, 16, ... 178, 32]

>>> train_labels[0]
1
```

- 실습 : 각자 데이터가 어떻게 생겼는지 확인해 보자
- 10,000개의 단어를 integer로 변환되어 들어가 있다. 변환된 integer의 최대값과 최소값은 무엇인가?
- 고급: 현재 숫자로만 나와있는데, 이를 원본의 영문으로 볼 수 있나?

(개별실습) 데이터 셋

- `word_index = imdb.get_word_index()`
- `word_index.items()` 는 무엇을 가지고 있나?
- `(word, index)`를 `(index, word)`로 변경할 수 있을까?
- 원본 영문으로 변환해 보도록 하시오

(개별실습) 데이터 셋

```
# word_index는 단어와 정수 인덱스를 매핑한 딕셔너리입니다
word_index = imdb.get_word_index()
# 정수 인덱스와 단어를 매핑하도록 뒤집습니다
reverse_word_index = dict([(value, key) for (key, value) in word_index.items()])
# 리뷰를 디코딩합니다.
# 0, 1, 2는 '패딩', '문서 시작', '사전에 없음'을 위한 인덱스이므로 3을 뺍니다
decoded_review = ' '.join([reverse_word_index.get(i - 3, '?') for i in train_data[0]])
```

- word_index = imdb.get_word_index() 는 무엇을 가지고 있나?
- word_index.items() 는 무엇을 가지고 있나?
- (word, index)를 (index, word)로 변경할 수 있을까?
- 원본 영문으로 변환해 보도록 하시오

(개별실습) 모델 변화

- 현재 은닉층을 2개 사용 (1개 사용한 경우, 3개 사용한 경우 성능 비교)
- 은닉층의 은닉 유닛의 개수 변화 (32개 unit, 64개 unit)
- loss 함수 변화 (binary crossentropy 대신 mse 사용)
- activation함수 변화 (relu 대신 tanh 사용)

Q. 뉴스 기사 토픽 분류

```
from keras.datasets import reuters  
  
(train_data, train_labels), (test_data, test_labels) = reuters.load_data(num_words=10000)
```

- 정답 코드를 참고하지 않고, 위의 라인과 이전 문제의 코드를 이용하여 학습을 시킬 수 있나?
- (제한시간: 30분)

Q. 집값 예측

```
from keras.datasets import boston_housing  
  
(train_data, train_targets), (test_data, test_targets) = boston_housing.load_data()
```