

# 혼자 공부하는 머신러닝 + 딥러닝



박해선 지음

---

## ★ ★ 혼자 공부하는 시리즈 소개

---

누구나 혼자 할 수 있습니다! 야심 찬 시작이 작심삼일이 되지 않도록 돕기 위해서 〈혼자 공부하는〉 시리즈를 만들었습니다. 낯선 용어와 친해져서 책장을 술술 넘기며 이해하는 것, 그래서 완독의 기쁨을 경험하고 다음 단계를 스스로 선택할 수 있게 되는 것이 목표입니다.

**지금 시작하세요.** 〈혼자 공부하는〉 사람들이 '때론 혼자, 때론 같이' 하며 힘이 되겠습니다.

## 1-2 코랩과 주피터 노트북

1. **답** ② 구글에서 제공하는 브라우저 기반의 파이썬 실행 환경은 코랩입니다.
  - ① 주피터 노트북이라고도 하며, 코랩과 유사하게 브라우저에서 실행 가능한 대화식 파이썬 실행 환경입니다. <https://jupyter.org>
  - ③ 구글에서 만든 웹 브라우저입니다. <https://www.google.co.kr/chrome>
  - ④ 파이썬과 R을 사용할 수 있는 통합 개발 환경(Integrated Development Environment, IDE)입니다. <https://www.anaconda.com>
2. **답** ④ 이외에도 \*혼공머신\*으로도 표현할 수 있습니다.
  - ① \*\*혼공머신\*\* : 굵게 표현합니다.
  - ② ~~혼공머신~~ : 취소선을 추가합니다.
  - ③ ‘혼공머신’ : 코드 서체로 씁니다.
3. **답** ③ 코랩은 구글 클라우드에서 실행됩니다.

## 1-3 마켓과 머신러닝

1. **답** ① 데이터를 표현하는 하나의 성질을 특성(feature)이라고 부릅니다.
2. **답** ④ k-최근접 이웃 알고리즘을 구현한 클래스는 KNeighborsClassifier입니다.
  - ① SGDClassifier는 경사 하강법을 사용한 분류 알고리즘을 구현한 클래스입니다. 경사 하강법은 4장에서 소개합니다.
  - ② LinearRegression은 선형 회귀 알고리즘을 구현한 클래스입니다. 선형 회귀는 3장에서 소개합니다.
  - ③ RandomForestClassifier는 트리 기반의 앙상블 알고리즘을 구현한 클래스입니다. 트리 모델은 5장에서 소개합니다.
3. **답** ② 사이킷런의 모델을 훈련할 때 사용하는 메서드는 fit()입니다.
  - ① predict() 메서드는 새로운 샘플에 대해 예측을 만듭니다.
  - ③ score() 메서드는 훈련한 모델을 평가합니다.
  - ④ transform() 메서드는 사이킷런의 전처리 클래스에서 데이터를 변환할 때 사용합니다. 이 메서드는 3장에서 소개합니다.

4.

```
kn = KNeighborsClassifier()
kn.fit(fish_data, fish_target)

for n in range(5, 50):
    # k-최근접 이웃 개수 설정
    kn.n_neighbors = n
    # 점수 계산
    score = kn.score(fish_data, fish_target)
    # 100% 정확도에 미치지 못하는 이웃 개수 출력
    if score < 1:
        print(n, score)
        break
```

## 2-1 훈련 세트와 테스트 세트

1. **답** ① 지도 학습은 샘플의 입력과 타깃이 준비되어 있을 때 사용할 수 있습니다.  
 ② 비지도 학습은 타깃이 없는 데이터에 적용하는 머신러닝 알고리즘입니다.  
 ③ 차원 축소는 비지도 학습의 하나로 데이터가 가지고 있는 특성의 개수를 줄이는 방법입니다.  
 비지도 학습과 차원 축소는 이 책의 2장에서 다룹니다.  
 ④ 강화 학습은 주어진 환경으로부터 보상을 받아 학습하는 머신러닝 알고리즘을 말합니다.
2. **답** ④ 샘플링 편향은 훈련 세트나 테스트 세트가 잘못 샘플링되어 전체 데이터를 대표하지 못하는 현상을 말합니다.
3. **답** ② 사이킷런은 입력 데이터에서 샘플이 행에 위치하고 특성이 열에 놓여 있다고 기대합니다.

## 2-2 데이터 전처리

1. **답** ③ 표준점수는 각 데이터가 0에서 몇 표준편차만큼 떨어져 있는지 나타내는 값입니다.  
 ② 원점수는 변환하지 않은 원래 점수를 말합니다.  
 ④ 사분위수는 데이터를 크기 순서대로 늘어 늘어놓았을 때 4등분 하는 수입니다.
2. **답** ① 테스트 세트는 반드시 훈련 세트의 통계 값으로 변환해야 합니다. 그렇지 않으면 훈련 세트에서 학습한 모델이 테스트 세트에서 올바르게 동작하지 않습니다.

### 3-1 k-최근접 이웃 회귀

1. ② k-최근접 이웃 회귀는 예측할 샘플에서 가장 가까운 k 개의 주변 샘플을 찾고 이 주변 샘플의 타깃값을 평균하여 예측값을 구합니다.

2.

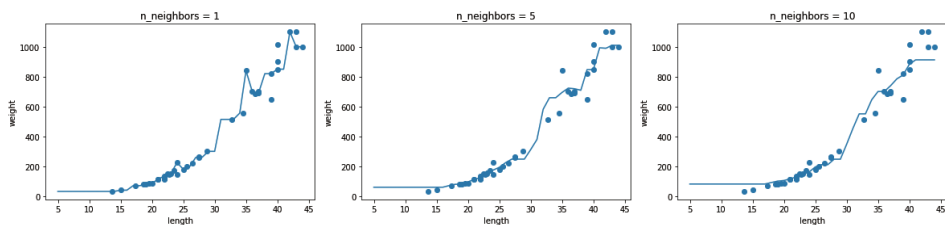
```
# k-최근접 이웃 회귀 객체를 만듭니다
knr = KNeighborsRegressor()

# 5에서 45까지 x 좌표를 만듭니다
x = np.arange(5, 45).reshape(-1, 1)

# n = 1, 5, 10일 때 예측 결과를 그래프로 그림니다
for n in [1, 5, 10]:
    # 모델을 훈련합니다
    knr.n_neighbors = n
    knr.fit(train_input, train_target)

    # 지정한 범위 x에 대한 예측을 구합니다
    prediction = knr.predict(x)

    # 훈련 세트와 예측 결과를 그래프로 그림니다
    plt.scatter(train_input, train_target)
    plt.plot(x, prediction)
    plt.title('n_neighbors = {}'.format(n))
    plt.xlabel('length')
    plt.ylabel('weight')
    plt.show()
```



### 3-2 선형 회귀

1. 답 ④ 모델 기반 학습에서 모델이 찾은 정보는 모델 파라미터에 저장됩니다. 선형 회귀에서는 방정식의 계수가 여기에 해당합니다.
2. 답 ① LinearRegression 클래스는 선형 회귀, 다항 회귀, 다중 회귀를 지원합니다.  
② PolynomialRegression이란 클래스는 없습니다.  
③ KNeighborsClassifier는 k-최근접 이웃 분류를 위한 클래스입니다.  
④ PolynomialClassifier란 클래스는 없습니다.

### 3-3 특성 공학과 규제

1. 답 ④ 최고 차수가 3이므로 추가되는 특성은 1, a, b, c, a2, b2, c2, ab, bc, ac, abc, ab2, ac2, bc2, ba2, ca2, cb2, a3, b3, c3입니다.
2. 답 ③ 특성을 표준화로 변환하는 전처리 클래스는 StandardScaler입니다.  
① Ridge는 릿지 회귀를 위한 클래스입니다.  
② Lasso는 라쏘 회귀를 위한 클래스입니다.  
④ LinearRegression은 선형 회귀를 위한 클래스입니다.
3. 답 ② 과대적합인 모델은 훈련 세트의 점수에 비해 테스트 세트의 점수가 크게 낮습니다.

### 4-1 로지스틱 회귀

1. 답 ② 2개보다 많은 클래스를 가진 분류 문제를 다중 분류 또는 다중 클래스 분류라고 부릅니다.  
① 이진 분류는 2개의 클래스, 즉 양성 클래스와 음성 클래스를 분류하는 문제입니다.  
③ 단변량 회귀는 하나의 출력을 예측하는 회귀 문제입니다.  
④ 다변량 회귀는 여러 개의 출력을 예측하는 회귀 문제입니다.
2. 답 ① 시그모이드 함수는 선형 방정식의 결과를 0과 1 사이로 압축하여 확률로 해석할 수 있습니다.  
② 소프트맥스 함수는 다중 분류에서 확률을 출력하기 위해 사용합니다.

3. 답 ③  $1 / (1 + e^{-0}) = 1 / (1 + 1) = 0.5$ 입니다. 따라서 이진 분류에서 `decision_function()`의 출력이 0보다 크면 시그모이드 함수의 값이 0.5보다 크므로 양성 클래스로 예측합니다.

#### 4-2 확률적 경사 하강법

1. 답 ② LinearRegression 클래스는 해석적인 방법으로 선형 방정식의 해를 구하기 때문에 특성의 스케일에 영향을 받지 않습니다.
- ① KNeighborsClassifier는 최근접 이웃을 찾기 위해 샘플 간의 거리를 계산합니다. 따라서 특성의 스케일이 다르면 잘못된 이웃을 선택할 수 있습니다.
  - ③ Ridge는 가중치를 규제하여 모델의 과대적합을 막습니다. 특성의 스케일이 다르면 이와 곱해지는 가중치의 스케일도 달라집니다. 이렇게 되면 큰 가중치에만 관심을 두게 되어 가중치를 공정하게 규제하지 못합니다.
  - ④ SGDClassifier는 손실 함수를 최소화하기 위해 가장 가파른 경로를 찾습니다. ③과 같이 특성의 스케일 때문에 가중치의 스케일에 차이가 크면 손실 함수를 최소화하는 경로를 올바르게 판단하지 못합니다.
2. 답 ③ 미니배치 경사 하강법은 훈련 세트에서 몇 개의 샘플(보통 2의 배수)을 뽑아 경사 하강법 알고리즘을 수행합니다.
- ① 확률적 경사 하강법은 훈련 세트에서 랜덤하게 1개의 샘플을 뽑아 경사 하강법 알고리즘을 수행합니다.
  - ② 배치 경사 하강법은 훈련 세트 전체를 사용해 경사 하강법 알고리즘을 수행합니다.

#### 5-1 결정 트리

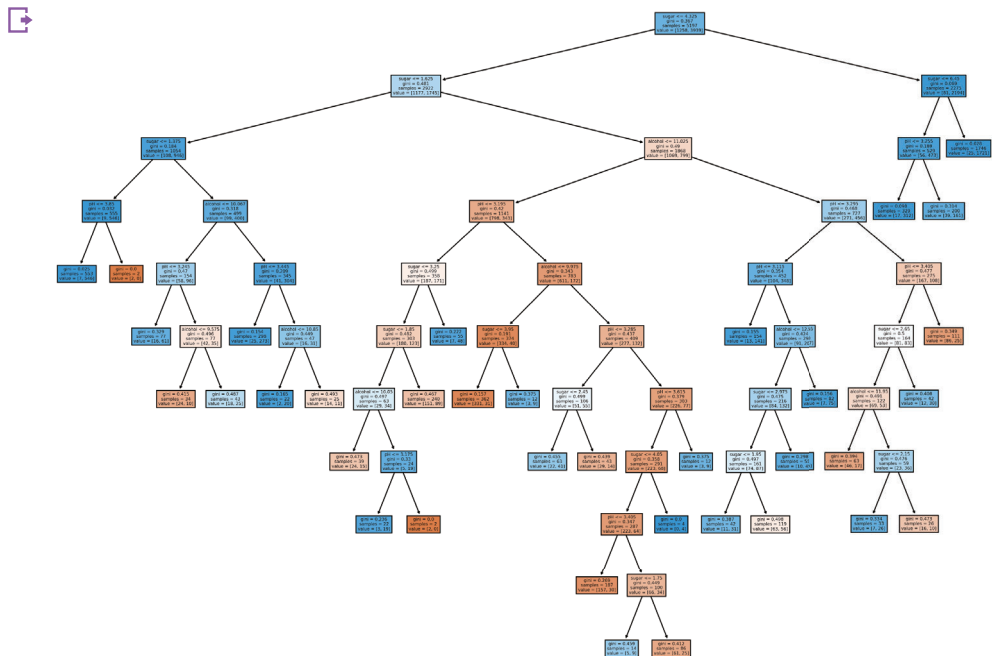
1. 답 ②, ④
- 지니 불순도 계산식 :  $1 - (\text{양성 클래스 비율}^2 + \text{음성 클래스 비율}^2)$   
 엔트로피 불순도 계산식 :  $-\text{음성 클래스 비율} \times \log_2(\text{음성 클래스 비율}) - \text{양성 클래스 비율} \times \log_2(\text{양성 클래스 비율})$
2. 답 ④ 결정 트리가 계산한 특성 중요도는 모델 객체의 `feature_importances_` 속성에 저장되어 있습니다.

3.

```
dt = DecisionTreeClassifier(min_impurity_decrease=0.0005, random_state=42)
dt.fit(train_input, train_target)
print(dt.score(train_input, train_target))
print(dt.score(test_input, test_target))
```

0.8874350586877044  
0.8615384615384616

```
plt.figure(figsize=(20,15), dpi=300)
plot_tree(dt, filled=True, feature_names=['alcohol', 'sugar', 'pH'])
plt.show()
```



## 5-2 교차 검증과 그리드 서치

1. **답** ① 교차 검증은 훈련 세트를 여러 개의 폴드로 나누고 하나의 폴드를 검증 세트로 두고 나머지 폴드를 훈련 세트로 사용합니다. 이런 방식으로 모든 폴드에 대해 반복합니다.
2. **답** ④ `train_test_split`은 데이터를 훈련 세트와 테스트로 분할합니다.  
 ① `cross_validate()`는 주어진 모델과 훈련 세트를 사용하여 기본 5-폴드 교차 검증을 수행합니다.  
 ② `GridSearchCV`와 ③ `RandomizedSearchCV`는 하이퍼파라미터 튜닝을 수행하면서 최상의 모델을 고르기 위해 교차 검증을 수행합니다.
3. 결정 트리의 노드를 랜덤하게 분할하기 때문에 100번의 반복에서 최적의 매개변수 조합을 찾지 못했습니다. 평균 검증 점수와 테스트 세트의 점수가 모두 조금 낮습니다.

```
gs = RandomizedSearchCV(DecisionTreeClassifier(splitter='random',
                                             random_state=42), params, n_iter=100,
                                             n_jobs=-1, random_state=42)
gs.fit(train_input, train_target)
print(gs.best_params_)
print(np.max(gs.cv_results_['mean_test_score']))
dt = gs.best_estimator_
print(dt.score(test_input, test_target))
```

```
{'max_depth': 43, 'min_impurity_decrease': 0.00011407982271508446,
 'min_samples_leaf': 19, 'min_samples_split': 18}
0.8458726956392981
0.786923076923077
```

## 5-3 트리의 앙상블

1. **답** ④ 앙상블 학습은 더 나은 성능을 내는 여러 개의 모델을 훈련하는 머신러닝 학습 방법입니다.
2. **답** ④ 이미지는 대표적인 비정형 데이터입니다.



① 엑셀 데이터, ② CSV 데이터, ③ 데이터베이스 데이터는 대표적인 정형 데이터입니다.

**3. 답** ① 랜덤 포레스트는 기본적으로 부트스트랩 샘플을 사용합니다.

② 엑스트라 트리의 bootstrap 매개변수의 기본값이 False이지만, True로 바꾸어 부트스트랩 샘플을 사용할 수도 있습니다.

③ 그레이디언트 부스팅과 ④ 히스토그램 기반 그레이디언트 부스팅은 부트스트랩 샘플을 사용하지 않습니다.

### 6-1 군집 알고리즘

**1. 답** ① hist() 함수는 첫 번째 매개변수에 입력한 배열 값의 히스토그램을 그립니다. bins 매개변수에 구간을 지정할 수 있으며 기본값은 10입니다.

② scatter()는 산점도를 그리는 함수입니다.

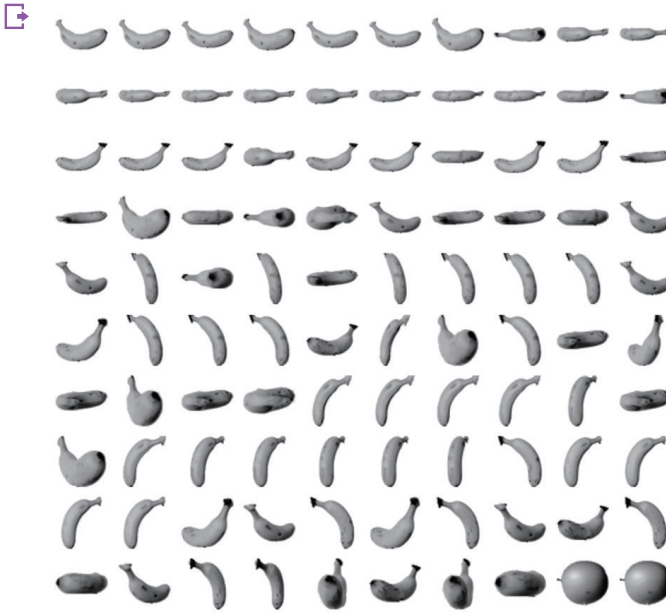
③ plot()은 선 그래프를 그리는 함수입니다.

④ bar()는 막대그래프를 그리는 함수입니다.

**2. 답** banana\_mean과 절댓값 오차가 가장 적은 100개의 사진을 고르면 2개를 제외하고 모두 바나나가 찾아집니다.

```
abs_diff = np.abs(fruits - banana_mean)
abs_mean = np.mean(abs_diff, axis=(1, 2))

banana_index = np.argsort(abs_mean)[:100]
fig, axs = plt.subplots(10, 10, figsize=(10, 10))
for i in range(10):
    for j in range(10):
        axs[i, j].imshow(fruits[banana_index[i*10 + j]], cmap='gray_r')
        axs[i, j].axis('off')
plt.show()
```



## 6-2 k-평균

1. **답** ④ 클러스터에 속한 샘플 개수는 클러스터 구성에 관련이 없습니다.
  - ① k-평균 알고리즘에서 클러스터에 속한 샘플을 평균 내어 클러스터 중심으로 정합니다.
  - ② 샘플은 가장 가까운 클러스터 중심에 속하며 이런 샘플들이 모여 하나의 클러스터를 구성합니다.
  - ③ 클러스터 중심을 센트로이드라고도 부릅니다.
2. **답** ① 엘보우 방법을 사용해 이너셔의 감소 정도가 꺾이는 클러스터 개수를 찾습니다.
  - ② 클러스터 개수가 많을수록 이너셔가 작게 나오기 때문에 무조건 작은 이너셔를 얻을 수 있는 클러스터 개수를 선택하면 안 됩니다.
  - ③ 군집은 타깃 없이 훈련하는 비지도 학습 알고리즘으로 대규모 데이터셋의 경우 직접 조사하여 몇 개의 클러스터가 만들어질지 파악하기 어렵습니다.
  - ④ 교차 검증은 지도 학습 모델이 훈련 데이터에서 얻을 수 있는 성능을 평가하는 도구입니다.

### 6-3 주성분 분석

1. 답 ② 일반적으로 특성의 개수만큼 주성분을 찾을 수 있습니다.
2. 답 ① (1000, 100) 크기 데이터셋에서 10개의 주성분을 찾아 변환하면 샘플의 개수는 그대로이고 특성 개수만 100에서 10으로 바뀝니다. 즉 (1000, 10)이 됩니다.
3. 답 ① 주성분 분석은 가장 분산이 큰 방향부터 순서대로 찾습니다. 따라서 첫 번째 주성분의 설명된 분산이 가장 큼니다.

### 7-1 인공 신경망

1. 답 ③ 밀집층에 있는 10개의 뉴런이 100개의 입력과 모두 연결되기 때문에 총  $100 \times 10 = 1,000$ 개의 가중치가 있고, 뉴런마다 1개의 절편이 있으므로 총 1,010개의 모델 파라미터가 있습니다.
2. 답 ② 이진 분류일 경우 출력층의 뉴런이 1개이고 선형 방정식의 결과를 확률로 바꾸기 위해 'sigmoid' 함수를 사용합니다.  
③ 'softmax' 함수는 다중 분류 신경망의 출력층에 사용합니다.  
④ 'relu' 함수는 이미지를 다루는 문제에서 자주 사용하는 활성화 함수입니다. 2절에서 자세히 소개합니다.  
① 'binary'라는 활성화 함수는 없습니다.
3. 답 ④ `compile()` 메서드의 `loss` 매개변수로 손실 함수를 지정하고 `metrics` 매개변수에서 측정하려는 지표를 지정할 수 있습니다.  
② `fit()` 메서드는 모델을 훈련하는 메서드입니다.  
① `configure()`와 ③ `set()` 메서드는 없습니다.
4. 답 ① 타깃값이 정수인 다중 분류일 경우 `compile()` 메서드의 `loss` 매개변수를 '`sparse_categorical_crossentropy`'로 지정합니다.  
② '`categorical_crossentropy`'는 타깃값이 원-핫 인코딩된 경우 사용합니다.  
③ '`binary_crossentropy`'는 이진 분류에 사용하는 손실 함수입니다.  
④ '`mean_square_error`'는 회귀 문제에 사용하는 손실 함수입니다.

## 7-2 심층 신경망

1. **답** ② 모델의 `add()` 메서드에는 층의 객체를 전달해야 합니다.  
 ①은 층의 클래스를 전달하고 있고, ③은 `Dense` 클래스의 매개변수를 `add()` 메서드에 전달합니다. ④는 `add()` 메서드에서 반환하는 값이 없기 때문에 함수 호출 오류가 발생합니다.
2. **답** ② 배치 차원을 제외한 입력의 차원을 일렬로 펼치려면 `Flatten` 클래스를 사용합니다.  
 ① `Plate`, ③ `Normalize` 클래스는 없습니다.  
 ④ `Dense` 층은 신경망에서 가장 기본적인 밀집층입니다. 입력의 차원을 변형하여 계산하지 않습니다.
3. **답** ③ '`relu`'는 이미지 처리 작업에 널리 사용되는 렐루 활성화 함수입니다.  
 ① '`linear`'는 선형 활성화 함수라고 부르며 실제로는 활성화 함수를 적용하지 않는다는 뜻입니다. 즉 뉴런의 선형 계산을 그대로 다음 층에 전달합니다. 일반적으로 '`linear`'는 회귀 작업을 위한 신경망의 출력층에 사용합니다.  
 ② '`sigmoid`'는 로지스틱 함수 또는 시그모이드 함수를 나타냅니다. 이 활성화 함수는 초창기 신경망에 많이 사용되었습니다.  
 ④ '`tanh`'는 하이퍼볼릭 탄젠트 함수를 나타냅니다. 순환 신경망에서 자주 사용됩니다.
4. **답** ① SGD는 기본 경사 하강법과 모멘텀, 네스테로프 모멘텀 알고리즘을 구현할 클래스입니다. 이런 알고리즘들은 모두 일정한 학습률을 사용합니다.  
 ② `Adagrad`, ③ `RMSprop`, ④ `Adam`은 모두 적응적 학습률 옵티마이저입니다.

## 7-3 신경망 모델 훈련

1. **답** ④ `fit()` 메서드에 검증 데이터를 전달하려면 `validation_data` 매개변수에 입력과 타깃을 튜플로 만들어 지정해야 합니다.
2. **답** ② `Dropout` 클래스에는 이전 층의 출력을 0으로 만들 비율을 지정합니다. 출력의 70%만 사용하려면 30%를 드롭아웃 합니다.
3. **답** ③ 모델 파라미터를 저장하는 메서드는 `save_weights()`입니다.  
 ① `save()` 메서드는 모델과 가중치를 모두 저장합니다.

- ② load\_model() 함수는 전체 모델을 읽어 들입니다.
- ④ load\_weights() 메서드는 파일에서 가중치를 읽습니다.

4. **답** ② 검증 손실을 지정하려면 monitor 매개변수를 'val\_loss' 설정합니다. 이 값이 monitor 매개변수의 기본값입니다.

- ① restore\_best\_weights의 매개변수를 지정하지 않았으므로 기본값 False가 적용되어 최상의 모델 파라미터를 복원하지 않습니다.
- ③ 'accuracy'는 훈련 세트의 정확도, ④ 'val\_accuracy'는 검증 세트의 정확도를 의미합니다.

## 8-1 합성곱 신경망의 구성 요소

1. **답** ③ (2, 2) 풀링은 특성 맵의 가로세로 크기를 절반으로 줄이기 때문에 합성곱의 출력 크기는 (8, 8, 5)가 됩니다. 세임 패딩이므로 합성곱 입력의 너비와 높이가 출력 크기와 같습니다. 또한, 컬러 이미지가므로 깊이(채널)는 3으로, 이 합성곱의 입력 크기는 (8, 8, 3)입니다.

2. 첫 번째 합성곱의 위치는 [[3, 0, 9], [5, 1, 2], [8, 2, 4]] 배열입니다. 여기에 [[2, 0, 1], [2, 0, 1], [2, 0, 1]]을 곱하면 47이 됩니다. 이런 식으로 가로세로 세 칸씩 이동하면서 입력과 커널을 곱하면 오른쪽과 같습니다.

47	8	42
41	12	38
43	14	46

3. 앞의 특성 맵에서 첫 번째 풀링의 위치는 [[6, 7], [1, 2]]입니다. 최대 풀링은 가장 큰 값을 고르는 것이므로 7이 됩니다. 이런 식으로 (2, 2) 영역이 겹치지 않게 이동하면서 최대 풀링을 계산하면 오른쪽과 같습니다.

10	8		
	7	7	10
9		8	5

## 8-2 합성곱 신경망을 사용한 이미지 분류

1. **답** ② strides에서 필터의 가로세로 이동 간격을 지정할 수 있습니다.

- ① kernel\_size는 필터의 가중치 가로세로 크기를 지정합니다.
- ③ padding은 합성곱 층의 패딩 타입을 지정합니다.
- ④ activation은 합성곱 출력에 적용할 활성화 함수를 지정합니다.

2. **답** ④ 'same'은 입력과 출력의 가로세로 크기가 같아지도록 입력에 알맞은 개수의 패딩을 추가합니다.
- ①, ② 'valid' 패딩은 입력에 패딩을 추가하지 않습니다.
3. **답** ④ MaxPooling2D의 풀링 크기는 2개의 정수(너비와 높이)로 구성해야 합니다.
- ① 풀링의 크기는 가로세로 크기가 같을 경우 하나의 정수로 지정할 수 있습니다.
- ② 풀링의 크기는 2개의 정수(너비와 높이)로 지정할 수 있습니다.
- ③ MaxPooling2D의 첫 번째 매개변수는 풀링의 크기이고, 두 번째 매개변수는 스트라이드 크기입니다.

### 8-3 합성곱 신경망의 시각화

1. **답** ② 필터가 원 모양을 따라 높은 값을 가지고 있으므로 동심원 패턴이 많은 이미지에서 가장 크게 활성화될 것입니다.
2. **답** ④ 케라스의 층 객체는 함수처럼 호출 할 수 있지만 모델 객체는 함수처럼 호출할 수 없습니다.
3. **답** ② Sequential 클래스의 layers[0]은 첫 번째 은닉층입니다. 첫 번째 은닉층의 출력은 두 번째 은닉층의 입력이며 모델의 입력이 아닙니다.
- ① Sequential 클래스의 layers[0]은 첫 번째 은닉층입니다. 첫 번째 은닉층의 입력이 모델의 입력이 됩니다.
- ③, ④ Sequential 클래스의 \_layers[0]은 InputLayer 객체입니다. 이 층의 입력과 출력은 같으며 모델의 입력을 나타냅니다.

### 9-1 순차 데이터와 순환 신경망

1. **답** ① 일반적으로 환자의 검사 데이터에는 순서가 없습니다. 예를 들어 체온, 심박수, 혈압 등은 순서가 없으며 독립적인 특성입니다. 하지만 만약 어떤 환자의 기록이 시간 별로 기록되어 있다면 순차 데이터로 다룰 수도 있습니다.
- ② 일정 기간 간격으로 기록된 데이터는 순차 데이터로 처리할 수 있습니다.
- ③ 태풍의 위도, 경도 위치가 일정 간격으로 기록되어 있으므로 순차 데이터입니다.

④ 악보의 음표는 순서대로 나열되어 있으므로 순차 데이터로 처리할 수 있습니다.

2. **답** ④ 순환 신경망에서는 순환층을 셀(cell)이라고도 부릅니다. 셀에서 출력되는 값을 은닉 상태라고 부릅니다.
3. **답** ② 셀의 은닉 상태 크기는 뉴런의 개수와 같습니다. 이 은닉 상태가 다음 타임스텝에 재사용될 때는 셀의 뉴런에 모두 완전히 연결됩니다. 따라서 필요한 가중치  $w_h$ 의 크기는 (셀의 뉴런 개수, 셀의 뉴런 개수) = (10, 10)이 됩니다.

### 9-2 순환 신경망으로 IMDB 리뷰 분류하기

1. **답** ② padding 매개변수가 'post'이므로 패딩은 항상 시퀀스의 끝에 추가되어야 합니다.
2. **답** ④ 케라스에서 제공하는 기본 순환층은 SimpleRNN입니다.
3. **답** ③ 입력 시퀀스에 있는 토큰 벡터의 크기가 10이고 순환층의 뉴런 개수가 16이므로  $w_x$ 의 크기는  $10 \times 16 = 160$ 개입니다. 순환층의 은닉 상태와 곱해지는  $w_h$ 의 크기는  $16 \times 16 = 256$ 개입니다. 마지막으로 뉴런마다 1개씩 총 16개의 절편이 있습니다. 따라서 이 순환층에 있는 모델 파라미터의 개수는  $160 + 256 + 16 = 432$ 개입니다.

### 9-3 LSTM과 GRU 셀

1. **답** ④ Conv2D는 합성곱 층 클래스입니다.
2. **답** ① LSTM에는 삭제 게이트, 입력 게이트, 출력 게이트가 있습니다.
3. **답** ② 순환층에서 모든 은닉 상태를 출력하려면 return\_sequences 매개변수를 True로 지정해야 합니다.