**1~2장**

|  |  |
| --- | --- |
| 1 | 난수로 초기화 |
| 2 | 전통적인 프로그래밍 : 데이터와 규칙을 입력하면 정답을 출력  머신러닝 : 데이터와 정답을 입력하면 규칙을 출력 |
| 3 | 1.SVM |
| 4 | RMSProp |
| 5 | 1.b, 2.a, 3.c |
| 6 | (128, 256, 256, 3) |

**3~4장**

|  |  |
| --- | --- |
| 1 | softmax |
| 2 | 매 훈련과 검증마다 다른 데이터 세트를 사용하기 때문에 과대적합을 방지할 수 있다. |
| 3 | [3, 5, 7] |
| 4 | 1 |
| 5 | model.add(layers.Dense(1)) |
| 6 | 1, 5 |

**5~6장**

|  |  |
| --- | --- |
| 1 | 최대풀링을 거친 데이터가 최대값으로 구성되어 있어 과대적합 가능성이 있다. |
| 2 | 이미지 증식 |
| 3 | 학습이 완료된 모델의 모든 층을 동결 시킨 후 상위 층 일부의 동결을 해제한 뒤 새로운 층과 함께 학습한다. |
| 4 | 5 |
| 5 | great라는 단어를 다른 단어들과 학습시키며 벡터로 표현해 임베딩 공간에 할당한다. |
| 6 | RNN에서 긴 시퀀스를 학습시킬 때 발생하는 그래디언트 소실 문제를 LSTM에서는 forget gate와 input gate를 통해 보완했기 때문. |
| 7 | 3 |
| 8 | 3 |