

딥 러닝을 이용한 자연어 처리 입문

11장 NLP를 위한 합성곱 신경망

발표자 : 김성윤

목차

1. 합성곱 신경망 (Convolution Neural Network)
2. 1D CNN
3. 1D CNN 실습
4. 문자 임베딩

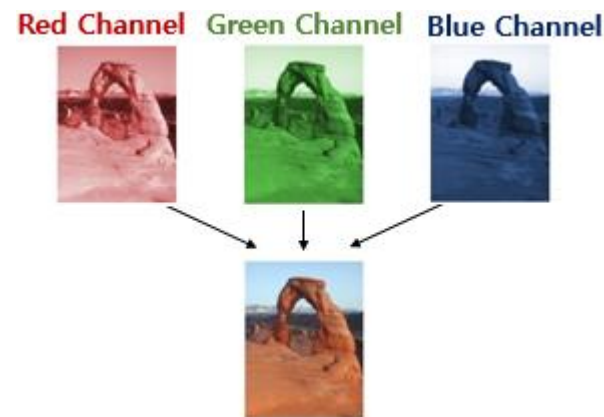
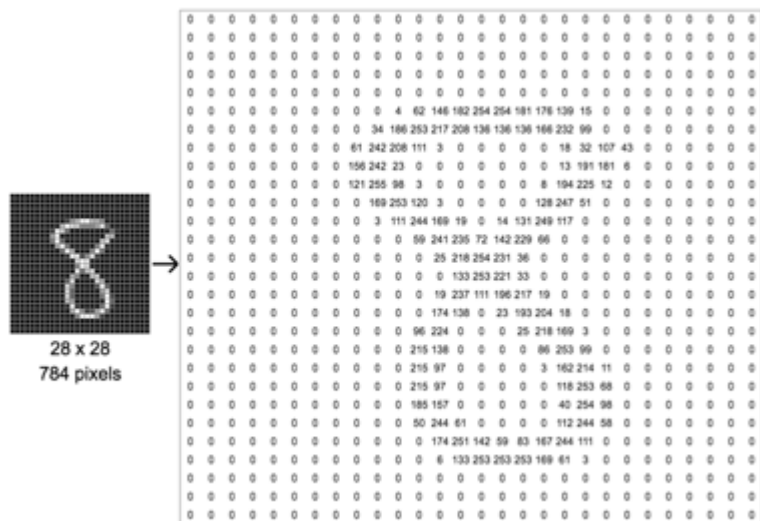
1. 합성곱 신경망 - 개념

- 합성곱층과 풀링층으로 구성
- 합성곱층 : (아래 그림에서) CONV(합성곱 연산) + RELU(활성화 함수) 층
- 풀링층 : 풀링 연산을 하는 구간
- 퍼셉트론의 공간적인 구조 정보가 유실되는 문제를 해결하기 위해 등장



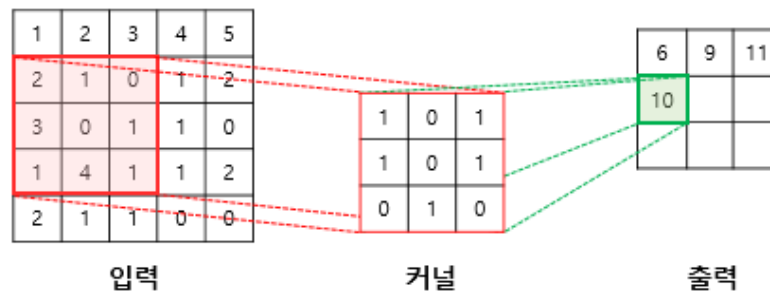
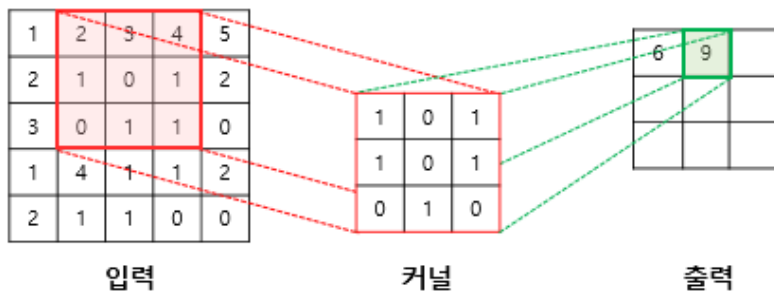
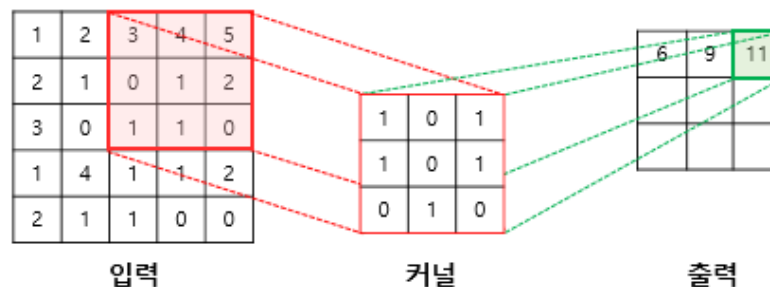
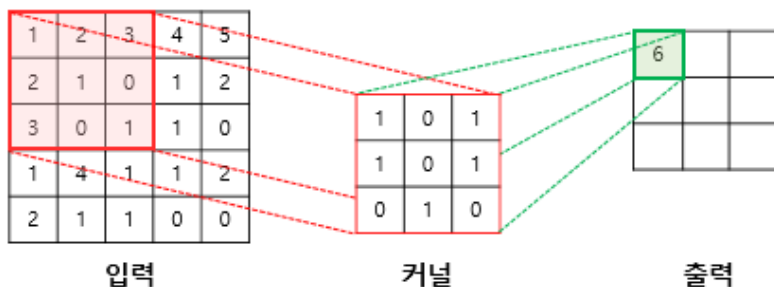
1. 합성곱 신경망 - 채널

- 이미지는 (높이, 너비, 채널(깊이))이라는 3차원 텐서로 구성
- 높이 : 세로 방향의 픽셀 수, 너비 : 가로 방향의 픽셀 수, 채널(깊이) : 색 성분
- 예) 흑백 사진의 채널 수 : 1
- 예) 컬러 사진의 채널 수 : 3 (RGB)



1. 합성곱 신경망 - 합성곱 연산

- 합성곱층 : 합성곱 연산을 통해 이미지 특징 추출
- 커널이 스트라이드(stride) 값만큼씩 이동하면서 합성곱 진행하며 특성 맵 제작
- 커널은 주로 3×3 또는 5×5 사용

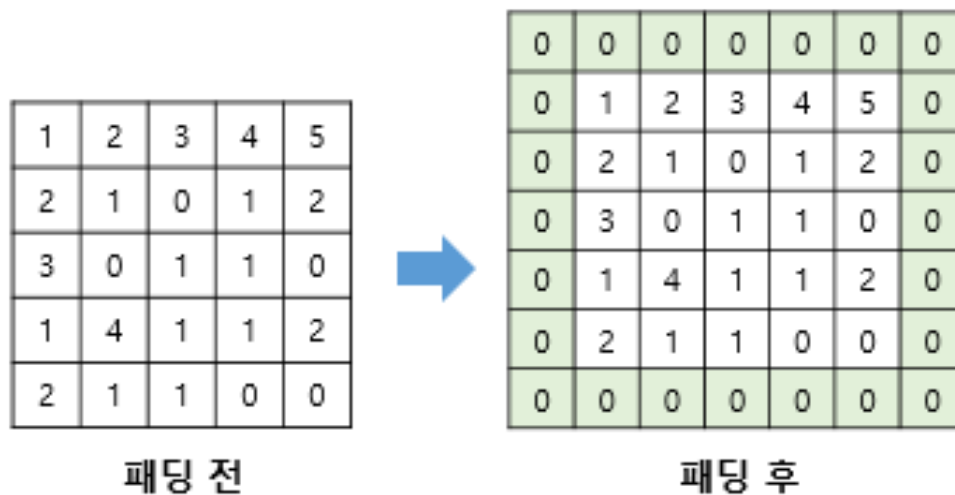


6	9	11
10	4	4
7	7	4

특성 맵(feature map)

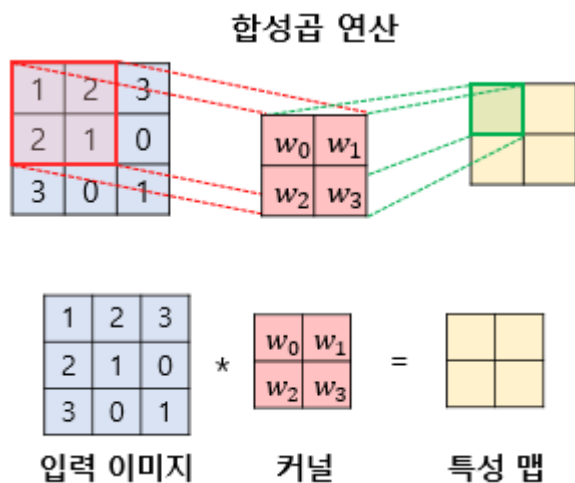
1. 합성곱 신경망 - 패딩

- 합성곱 연산 후, 특성 맵의 크기가 입력의 크기에 비해 작아지지 않고 싶을 때 사용

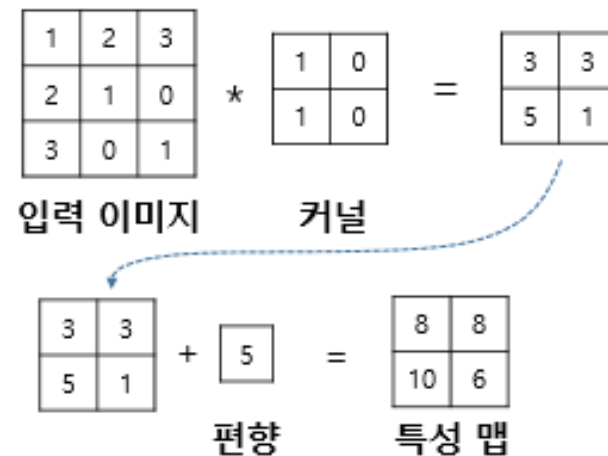
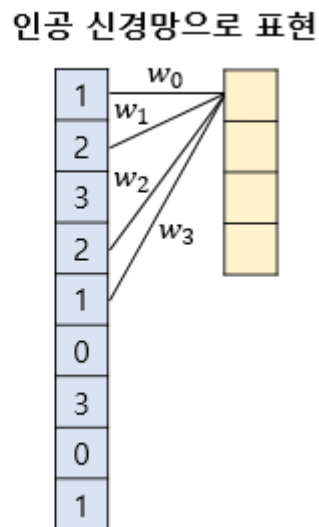


1. 합성곱 신경망 – 가중치 & 편향

- 이미지 전체를 훑으면서 사용하는 가중치는 w_0, w_1, w_2, w_3 4가지만 사용
- 다층 퍼셉트론에 비해 적은 가중치 사용



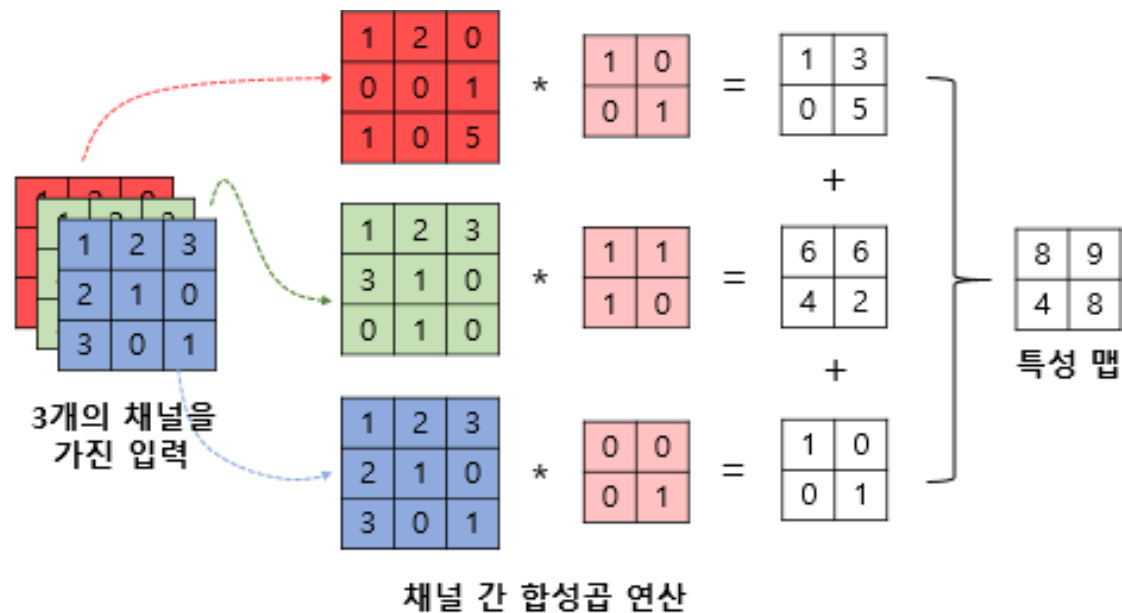
합성곱에서의 가중치



합성곱에서의 편향

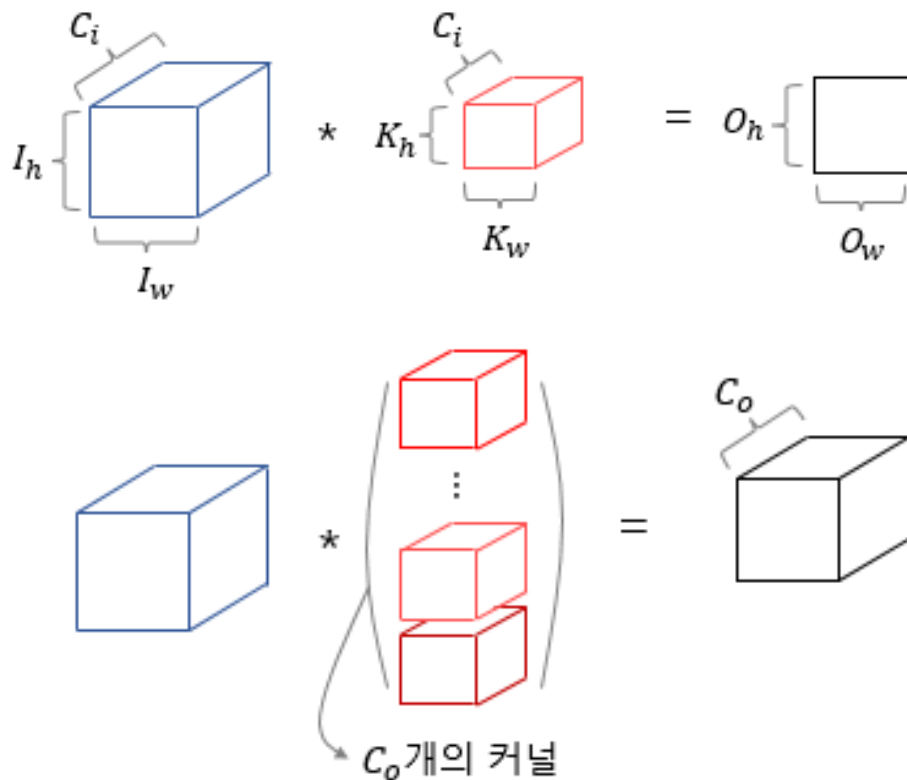
1. 합성곱 신경망 - 다수의 채널 입력

- 입력 데이터의 채널 수 = 커널의 채널 수
- 채널마다 합성곱 연산 수행 후, 모두 합하여 특성 맵을 얻음.
- 주의) 3개의 채널을 가진 1개의 커널 (O), 3개의 커널 (X)



1. 합성곱 신경망 – 3차원 텐서의 합성곱

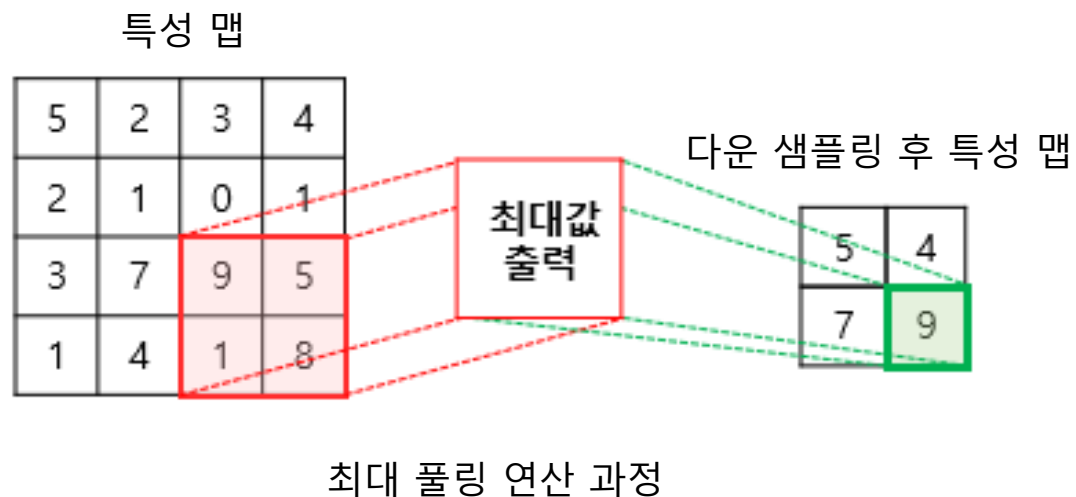
- 가중치 매개변수의 총 수 : $K_i \times K_o \times C_i \times C_o$



- I_h : 입력의 높이
- I_w : 입력의 너비
- K_h : 커널의 높이
- K_w : 커널의 너비
- O_h : 특성 맵의 높이
- O_w : 특성 맵의 너비
- C_i : 입력 데이터의 채널

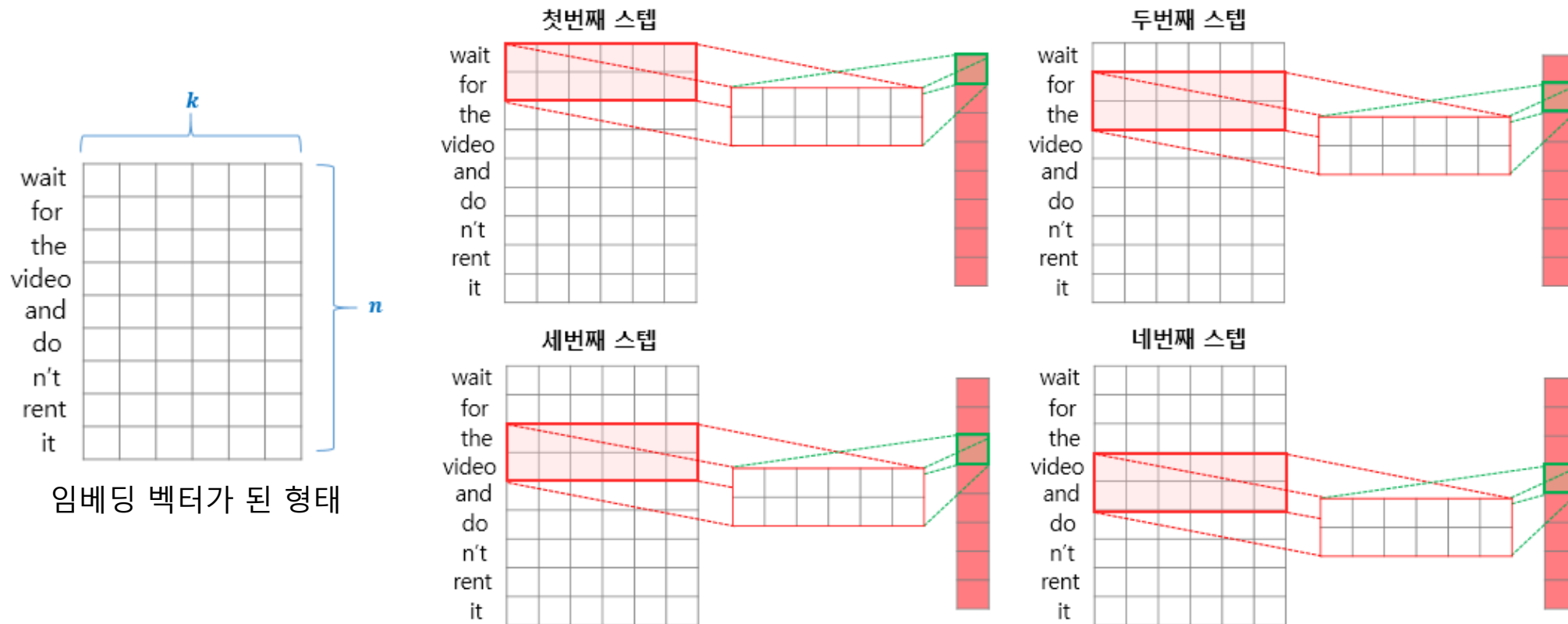
1. 합성곱 신경망 - 풀링

- 풀링 층 : 특성 맵을 다운샘플링하여 특성 맵의 크기를 줄이는 풀링 연산이 이루어짐
- 일반적으로 **최대 풀링** & **평균 풀링** 사용
- 합성곱 연산과 달리 채널 수가 변하지 않음



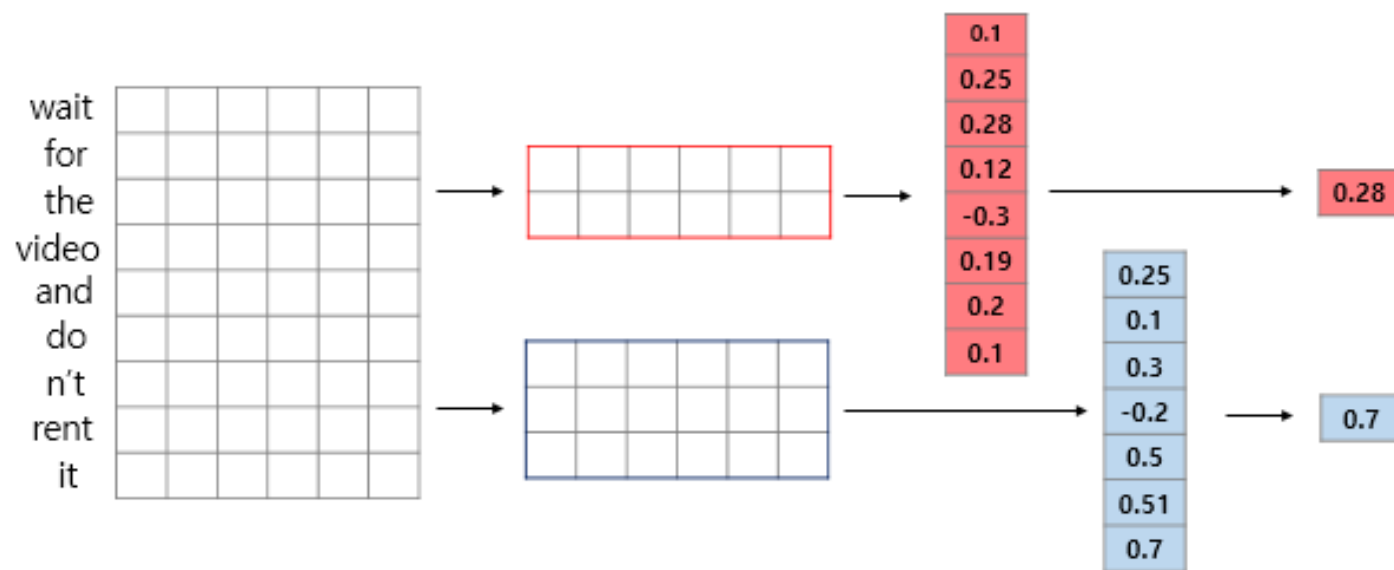
2. 1D CNN – 개념

- 각 문장은 임베딩 층을 지나 각 단어가 임베딩된 상태로 LSTM 입력이 됨
- 커널의 크기에 따라 단어의 묶음의 크기가 달라짐



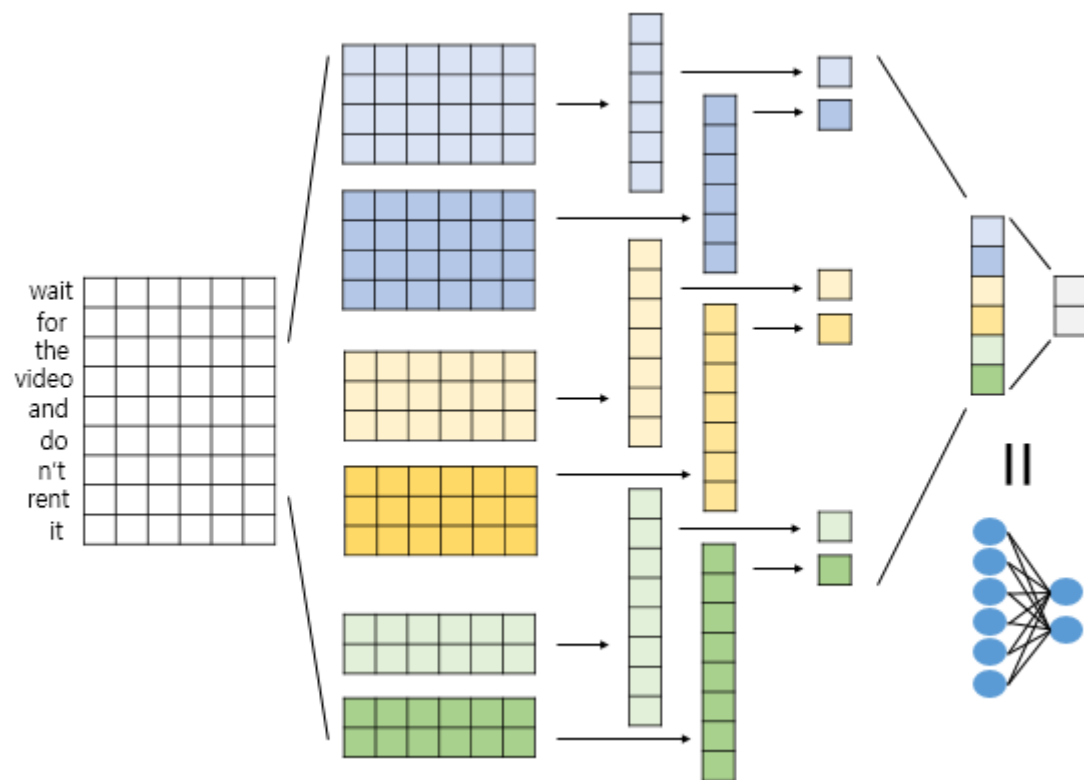
2. 1D CNN – 맥스 풀링

- 맥스 풀링 : 합성곱 연산에서 얻은 벡터의 스칼라 값 중 가장 큰 값을 빼는 것



2. 1D CNN – 신경망 설계

- 뉴런이 2개인 출력층에 완전 연결하므로 텍스트 분류를 수행 (소프트맥스 함수 사용)



2. 1D CNN – 케라스 이용

```
from tensorflow.keras.layers import Conv1D, GlobalMaxPooling1D

model = Sequential()
model.add(Conv1D(num_filters, kernel_size, padding='valid', activation='relu'))
```

합성곱 층 추가

```
model = Sequential()
model.add(Conv1D(num_filters, kernel_size, padding='valid', activation='relu'))
model.add(GlobalMaxPooling1D())
```

풀링 층 추가

3. 1D CNN 실습 – IMDB 리뷰 & 스팸 메일

```
model = Sequential()  
model.add(Embedding(vocab_size, embedding_dim))  
model.add(Dropout(dropout_ratio))  
model.add(Conv1D(num_filters, kernel_size, padding='valid', activation='relu'))  
model.add(GlobalMaxPooling1D())  
model.add(Dense(hidden_units, activation='relu'))  
model.add(Dropout(dropout_ratio))  
model.add(Dense(1, activation='sigmoid'))
```

Conv1D() 함수를 이용한 층 생성

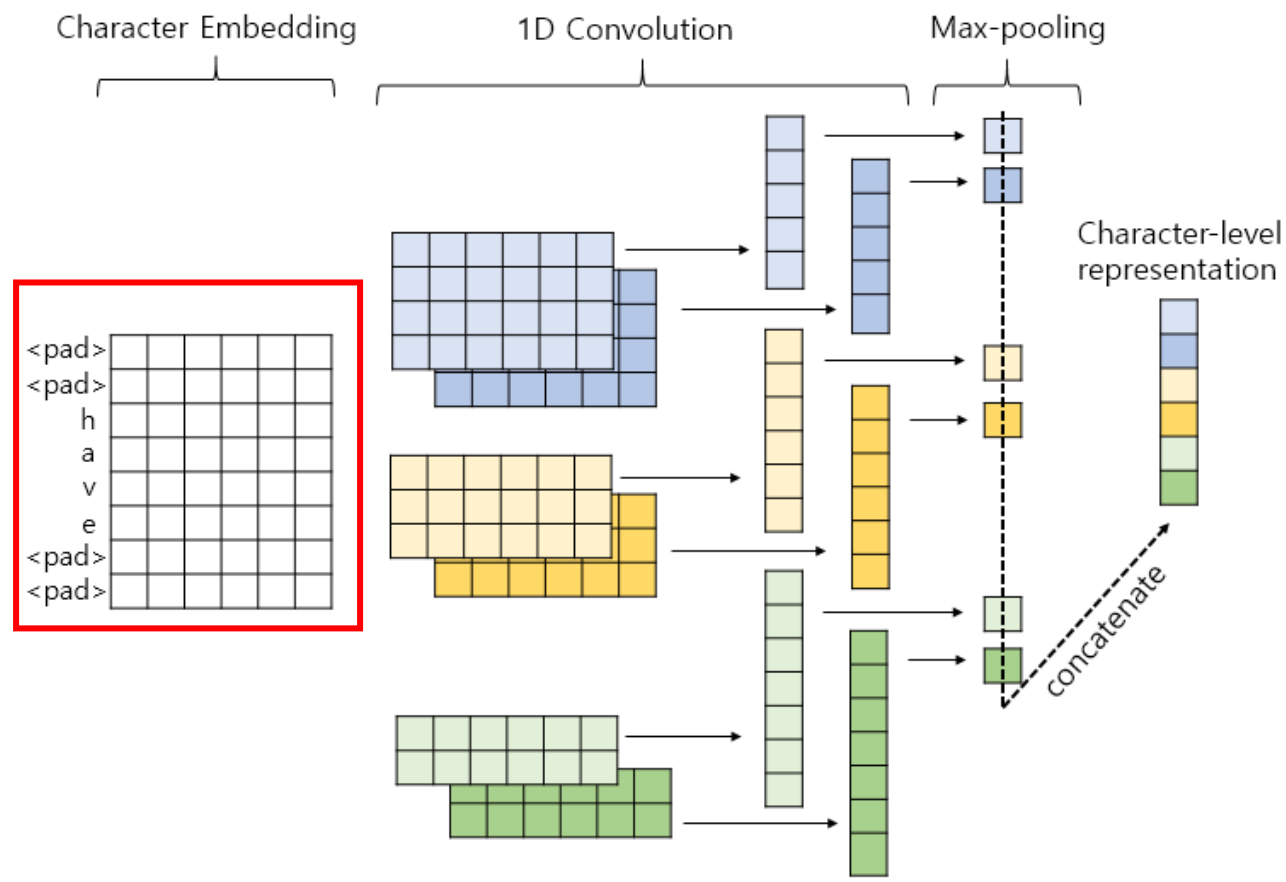
3. 1D CNN 실습 – Multi-Kernel 1D CNN

```
conv_blocks = []  
  
for sz in [3, 4, 5]:  
    conv = Conv1D(filters = num_filters,  
                  kernel_size = sz,  
                  padding = "valid",  
                  activation = "relu",  
                  strides = 1)(z)  
  
    conv = GlobalMaxPooling1D()(conv)  
    conv_blocks.append(conv)
```

```
z = Concatenate()(conv_blocks) if len(conv_blocks) > 1 else conv_blocks[0]  
z = Dropout(dropout_ratio[1])(z)  
z = Dense(hidden_units, activation="relu")(z)  
model_output = Dense(1, activation="sigmoid")(z)
```


5. 문자 임베딩 - 1D CNN

- 사람이 모르는 단어를 더 작게 쪼개서 뜻을 이해하는 능력을 흉내낸 것
- 예) misunderstand = 'mis-' + 'understand'



5. 문자 임베딩 – BiLSTM

- 순방향의 마지막 시점의 은닉 상태와 역방향의 첫 번째 시점의 은닉 상태를 연결

