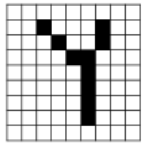


CNN in NLP

합성곱 신경망(Convolution Neural Network, CNN)

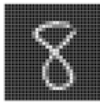
CNN은 **필터링 기법**을 인공지능망에 적용하여 **이미지**를 효과적으로 처리할 수 있는 심층 신경망 기법으로 행렬로 표현된 필터의 각 요소가 데이터 처리에 적합하도록 자동으로 학습되는 과정을 통해 이미지를 분류하는 기법



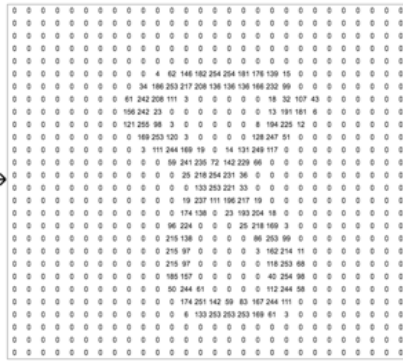
↓ 변환



기존 다층 퍼셉트론(MLP)으로 이미지 처리시, 1차원 텐서인 벡터로 변환해 사용.
이때 공간적 구조 정보는 유실됨.



28 x 28
784 pixels



이미지를 3차원 텐서로 표현

이미지는 (높이, 너비, **채널**)이라는 3차원 텐서로, 높이는 이미지의 세로 방향 픽셀 수, 너비는 이미지의 가로 방향 픽셀 수, 채널은 색 성분을 의미

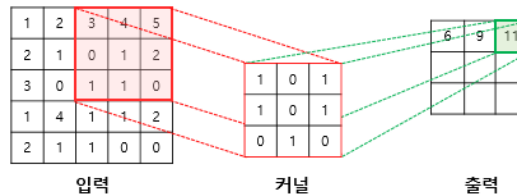
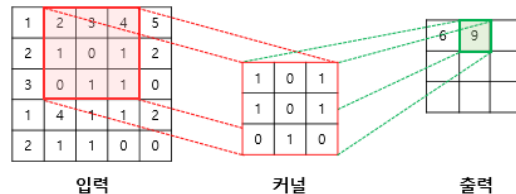
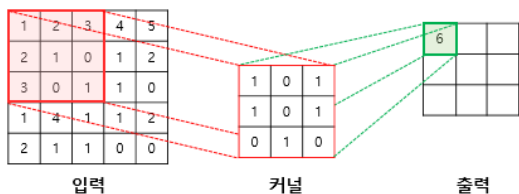
합성곱 신경망(Convolution Neural Network, CNN)

합성곱 연산(Convolution operation)

합성곱층(convolution layer)은 합성곱 연산을 통해서 **이미지의 특징을 추출**하는 역할

커널(kernel)(또는 필터(filter))라는 $n \times m$ 크기의 행렬로 (높이 \times 너비) 크기의 이미지를 처음부터 끝까지 겹치며 훑으면서 $n \times m$ 크기의 겹쳐지는 부분의 각 이미지와 커널의 원소의 값을 곱해서 모두 더한 값을 출력

커널은 일반적으로 3×3 , 5×5 사용



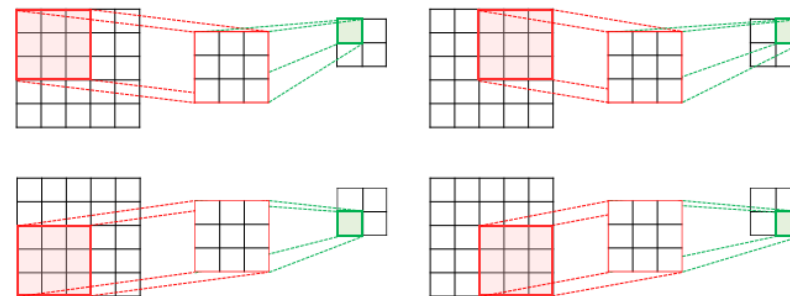
<https://stanford.edu/~shervine/l/ko/teaching/cs-230/cheatsheet-convolutional-neural-networks>

6	9	11
10	4	4
7	7	4

특성 맵(feature map)

커널을 사용하여 합성곱 연산을 통해 나온 결과를 **특성 맵(feature map)**이라고 함

단, 이때 이동 범위를 **스트라이드(stride)**라 하며, 예에서는 1임.

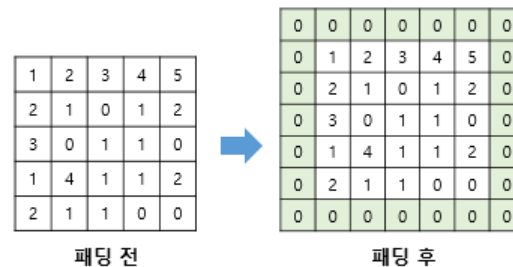


스트라이드(stride)=2인 경우

합성곱 신경망(Convolution Neural Network, CNN)

패딩(Padding)

- 합성곱 연산 이후에도 특성 맵의 크기가 입력의 크기와 동일하게 유지되도록 하고 싶다면 **패딩(padding)**을 사용
- 패딩이란, 지정된 개수의 폭만큼 테두리를 추가하는 것



제로 패딩(zero padding)

$$O_h = \text{floor}(\frac{I_h - K_h + 2P}{S} + 1)$$

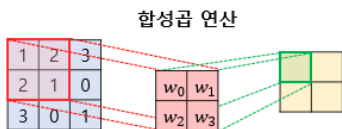
$$O_w = \text{floor}(\frac{I_w - K_w + 2P}{S} + 1)$$

특성맵 크기를 수식으로 보면 아래와 같음.

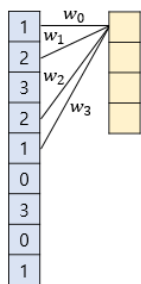
O_h, O_w는 특성 맵의 높이 및 너비, I는 입력, K는 커널, P는 패딩, S는 스트라이드

가중치

- 합성곱 신경망에서 가중치는 **커널 행렬의 원소들**
- 합성곱 연산을 통해서 특성 맵을 얻고, 활성화 함수를 지나는 연산을 하는 층을 합성곱 층(convolution layer)이라고 함

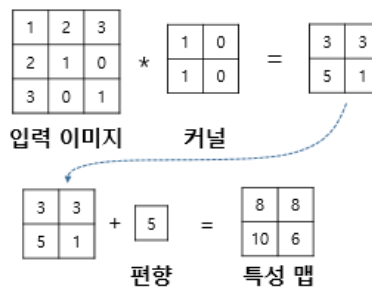


인공 신경망으로 표현



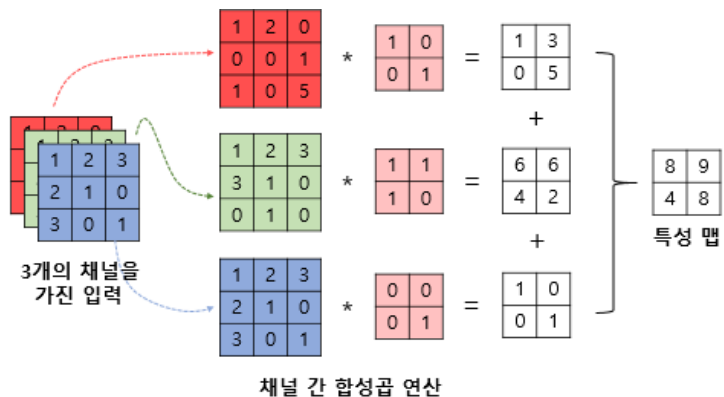
편향

- 합성곱 신경망에도 편향(bias)를 당연히 추가할 수 있음
- 편향은 하나의 값만 존재하며, 커널이 적용된 결과의 모든 원소에 더해짐



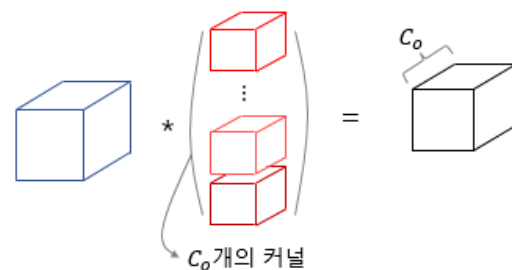
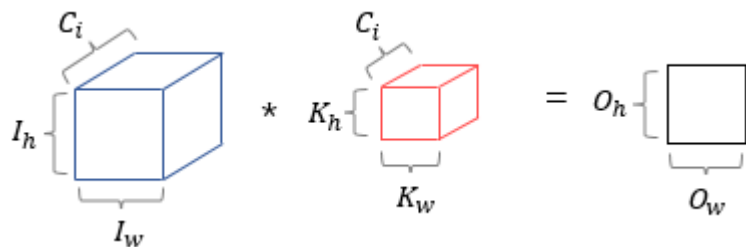
합성곱 신경망(Convolution Neural Network, CNN)

다수의 채널을 가질 경우의 합성곱 연산(3차원 텐서의 합성곱 연산)



실제로 합성곱 연산의 입력은 '다수의 채널을 가진' 이미지 또는 이전 연산의 결과로 나온 특성 맵일 수 있음

만약, 다수의 채널을 가진 입력 데이터를 가지고 합성곱 연산을 한다고 하면 커널의 채널 수도 입력의 채널 수만큼 존재해야 함



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

합성곱 신경망(Convolution Neural Network, CNN)

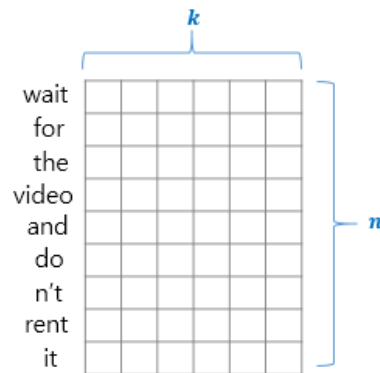
풀링(Pooling)

합성곱 층(합성곱 연산 + 활성화 함수) 다음에는 풀링 층을 추가하는 것이 일반적임.
풀링 층에서는 **특성 맵을 다운샘플링(downsampling)**하여 특성 맵의 크기를 줄이는 풀링 연산이 이루어짐.
풀링 연산에는 일반적으로 최대 풀링(max pooling)과 평균 풀링(average pooling)이 사용됨.

유형	최대 풀링	평균 풀링
목적	각 풀링 작업은 현재 뷰(view)에서 최대 값을 선택합니다.	각 풀링 작업은 현재 뷰(view)의 값들의 평균을 취합니다.
그림		
코멘트	<ul style="list-style-type: none">• 탐지된 특징을 보존• 가장 보편적으로 사용	<ul style="list-style-type: none">• 피쳐맵의 다운샘플링• LeNet에 사용

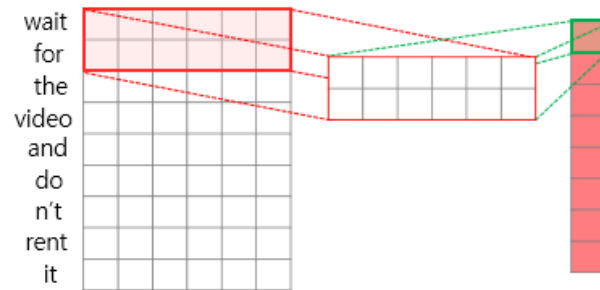
자연어 처리를 위한 1D CNN

문장이 토큰화, 패딩, 임베딩 층(Embedding layer)을 거친다면 다음과 같은 문장 형태의 행렬로 변환



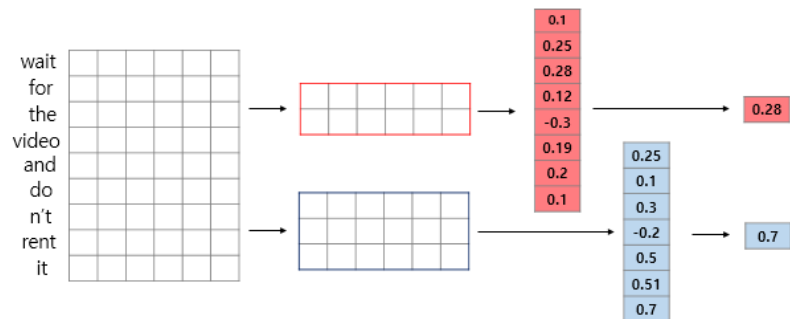
1D 합성곱(1D Convolutions)

- 1D 합성곱 연산에서 **커널의 너비**는 문장 행렬에서의 **임베딩 벡터의 차원**과 동일하게 설정
- 그렇기 때문에 1D 합성곱 연산에서는 **커널의 높이**만으로 해당 **커널의 크기**라고 간주
- 따라서 1D 합성곱 연산에서는 커널이 문장 행렬의 높이 방향으로만 움직임.
- CNN에서의 커널은 신경망 관점에서는 가중치 행렬이므로 커널의 크기에 따라 학습하게 되는 파라미터의 수는 달라짐.
- 1D 합성곱 연산과 자연어 처리 관점에서는 커널의 크기에 따라서 **참고하는 단어의 묶음의 크기**가 달라짐 (**n-gram**이 달라짐)



자연어 처리를 위한 1D CNN

맥스 풀링: 각 합성곱 연산으로부터 얻은 결과 벡터에서 가장 큰 값을 가진 스칼라 값을 빼냄

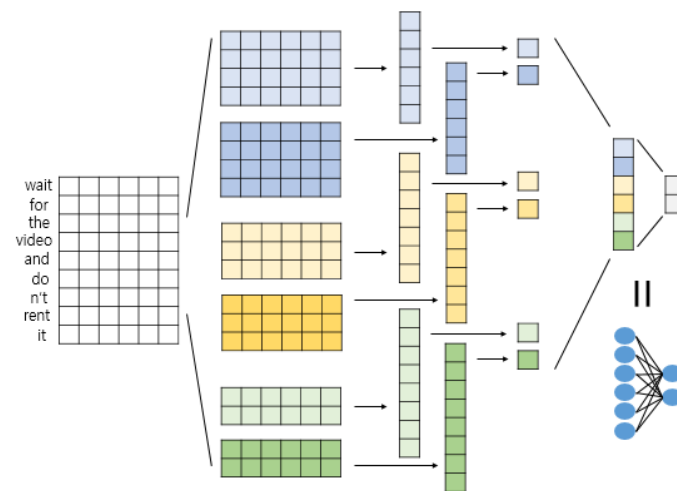


신경망 설계

소프트맥스 함수를 사용할 것이므로 출력층에서 뉴런의 개수가 2인 신경망을 설계

크기가 4인 커널 2개, 3인 커널 2개, 2인 커널 2개를 사용

나온 벡터 6개 → 맥스 풀링 후 6개 스칼라 값 → 전부 연결(concatenate) 후 하나의 벡터 (=1D CNN을 통해서 문장으로부터 얻은 벡터) → 뉴런이 2개인 출력층에 완전 연결(Dense layer를 사용)



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

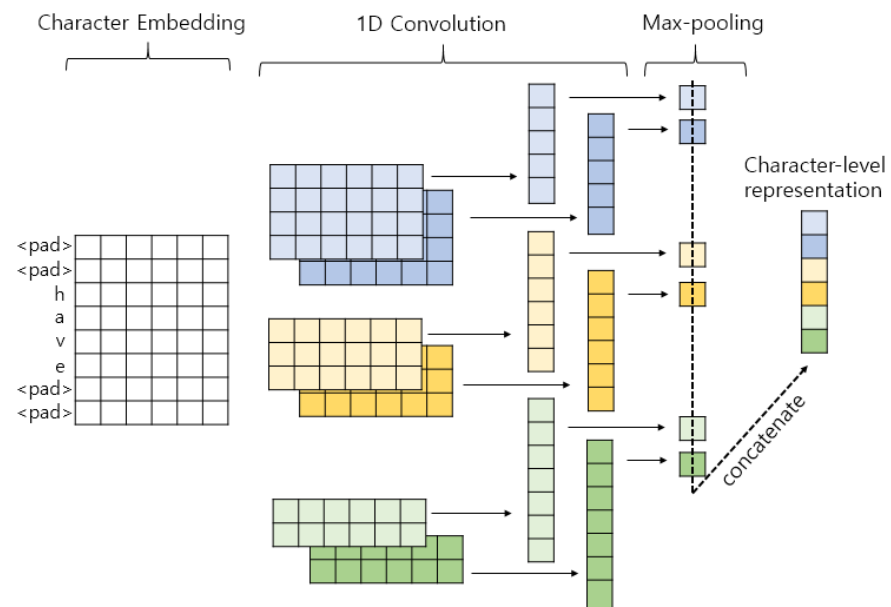

문자 임베딩 (Character Embedding)

문자 임베딩(Character Embedding): 워드 임베딩과는 다른 방법으로 단어의 벡터 표현 방법을 얻음

1D CNN을 사용한 문자 임베딩

단어 'have' → 'h', 'a', 'v', 'e'와 같이 문자 단위
→ 1d cnn으로 하나의 벡터로 만듦

최종적으로 이렇게 얻은 벡터를 단어 'have'의 벡터로 사용



참고자료

- 딥 러닝을 이용한 자연어 처리 입문(<https://wikidocs.net/book/2155>)
- 합성곱 신경망(<https://stanford.edu/~shervine/l/ko/teaching/cs-230/cheatsheet-convolutional-neural-networks>)