CNN을 이용한 텍스트 분류

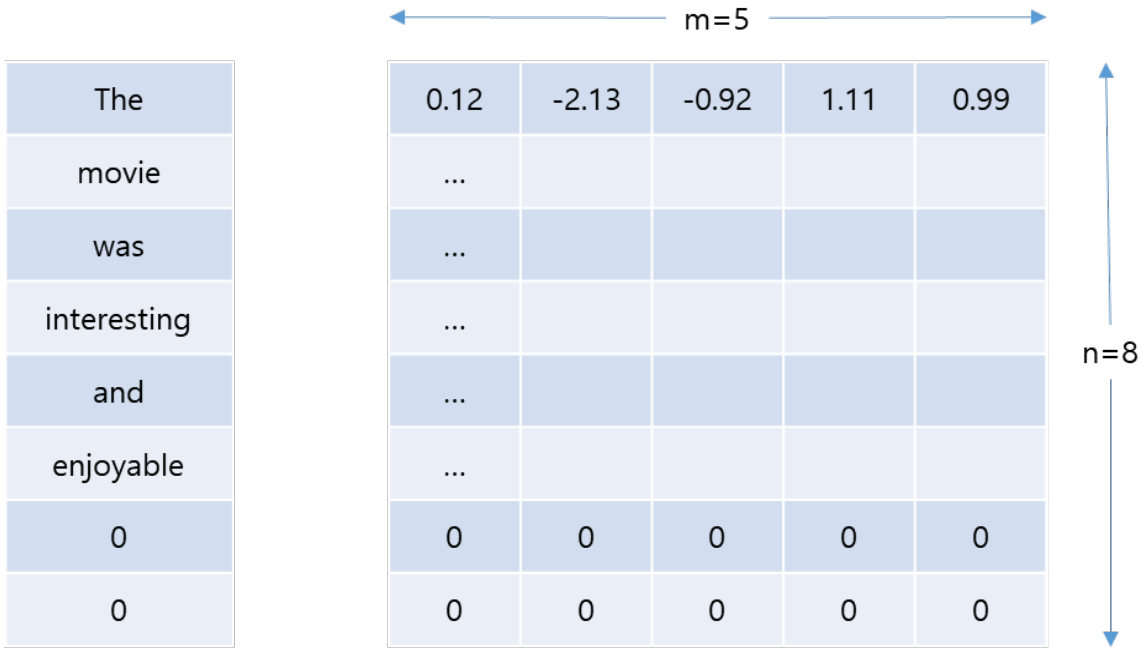
Sang Yup Lee

CNN을 이용한 텍스트 분류

- 순서 1
  - 문서를 3D 형태, 즉 (n, m, c) 로 표현 (즉, 하나의 이미지와 같은 형태로 표현)
    - n은 문서를 표현할 때 사용하는 최대 단어 수
    - m 은 한 단어를 표현하는 임베딩 벡터의 차원
    - c는 이미지 데이터에서의 채널수로 문서에서는 1

CNN을 이용한 텍스트 분류

- 예) n=8, m=5, c=1
  - 문서: The movie was interesting and enjoyable



CNN for text classification

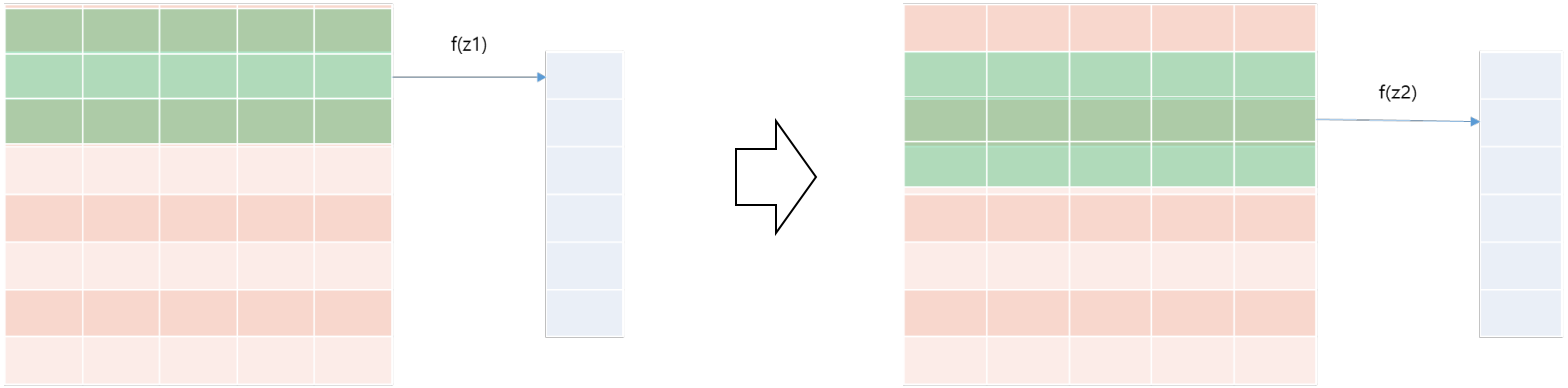
CNN을 이용한 텍스트 분류

- 순서 2
  - 합성곱 필터 적용하기
    - 텍스트의 경우, nxmxc 의 문서에 대해서 kxmxc 형태의 필터를 적용
      - 여기에서 k < m
      - 앞의 예에서는 kx5x1 를 적용
      - k=3인 경우

w1,1	w1,2	w1,3	w1,4	w1,5
w2,1	w2,2	w2,3	w2,4	w2,5
w3,1	w3,2	w3,3	w3,4	w3,5

CNN을 이용한 텍스트 분류

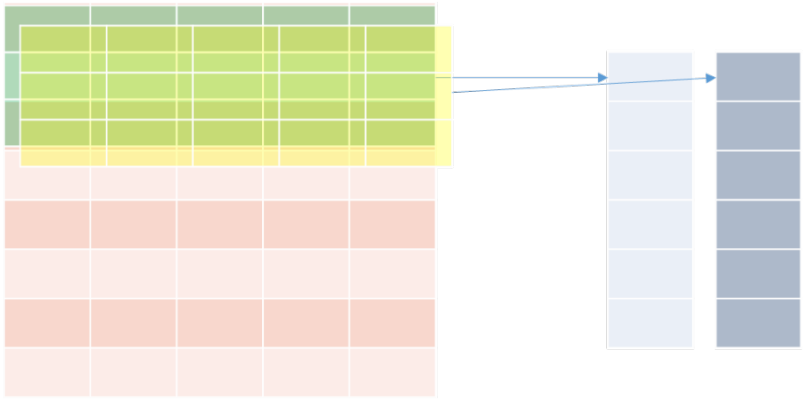
- 필터의 적용
  - 문서의 경우는 보통 stride=1



결과물은 (n-k+1) 크기의 1D array

CNN을 이용한 텍스트 분류

- 필터의 적용 (cont'd)
  - 크기가 같은 필터 2개를 적용하는 경우



nxm 형태의 문서 array에 kxm 형태의 필터를 h개 적용하는 경우에 생기는 결과물은 (n-k+1)xh

CNN을 이용한 텍스트 분류

- 그 다음 순서
  - 이미지에서와 마찬가지로 pooling layer 추가
  - Flattening
  - Fully connected layer
  - 출력측 with softmax()
- Python coding
  - See "CNN_imdb_example.ipynb"