



lecture 11 preview

1. Intro to CNNs

- RNN의 문제점
 - prefix context를 모두 포함 - 문장을 나누지 못하고 전체에 대해 계산해야한다.
 - last hidden state에 정보가 축약되어있다. 이전에 대한 정보들은 희미해진다. GRU, Attention 등을 해결책으로 내세움.
- CNN : 특정길이로 subsequence 분할 후 특징을 추출한다. → 문법적으로 맞는지 확인이 불가능하다.
 - Convolution 연산 : filter사이즈만큼의 영역 의 닷 프로덕트를 구한다. stride만큼 이동하여 닷프로덕트를 구하는 것을 반복한다.

2. 1d convolution for text

- 필터가 한 방향으로만 이동한다.
- 주로 embedded vector 사용
- 원래의 시퀀스의 길이를 유지하기 위해 위아래로 패딩을 준다.
- 필터를 여러 개 두면 output의 차원이 늘어난다. = 기존 문장에 대한 정보량이 늘어난다.
- Max pooling : 뽑아온 정보들을 요약하고 싶을 때 사용한다. 필터의 feature에서 가장 큰 값 하나를 뽑는다.
 - k-max pooling : k개씩 뽑는다. 나온 순서를 보존하여 적어준다.
- Average pooling : feature들의 평균을 취한다. NLP에서는 정보들이 균일하지 않고 sparse하기에 Max pooling을 더 선호한다.
- local max pooling : vision영역에서 사용, NLP에서는 잘 사용하지 않음.
 - stride : 건너뛰는 칸의 크기
 - dilation rate : 커널 사이의 간격(시야)

3. CNN for classification

- 단어들을 matrix 아니라 concat하여 연산
- gated units
 - residual block
 - highway block
 - x에 대한 정보를 얼마나 넘길 것인지에 따라서 구분, 같은 역할
- batch norm
 - CNN에 사용된다.
 - convolution의 output을 배치 별로 평균0, 분산1로 정규화시킨다.
 - 모델의 안정적 학습을 가능하게함
 - 배치 별 업데이트는 변동이 클 수 있어 많이 사용하면 안됨
 - parameter initialization에 덜 민감하다
 - learning rate tuning이 쉬워진다.
- 1x1 convolutions
 - 적은 파라미터로 채널 축소가 가능하다
 - fully connected layer의 인풋값으로 사용 가능
- CNN for NLP
 - 구현이 잘 되어있고 사용하기 편함
 - 첫번째 단어나 마지막 단어에 중요한 정보가 있으면 이것을 포착하기 위해 convolution layer가 필요하다
 - 전체를 고려하는 RN과 로컬한 정보만을 고려하는 CNN의 장점을 합친 것이 self attention

4. Deep CNN for NLP

- VD-CNN : 층이 깊어질수록 결과가 좋다