Seq2Seq

Introduction

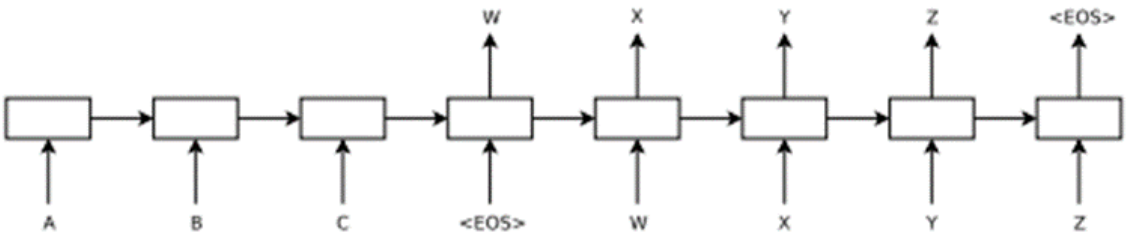
DNN (Deep Neural Network)는 음성 인식, 사물 인식 등에서 꾸준한 성과를 내어왔다. 하지만 input size가 fixed된다는 한계점이 존재하기 때문에 sequencial problem을 제대로 해결할 수 없다는 한계점이 존재했다.

본 논문에서는 2개의 LSTM (Long Short Term Memory)을 각각 Encoder, Decoder로 사용해 sequencial problem을 해결하고자 했다. 이를 통해 많은 성능 향상을 이루어냈으며, 특히나 long sentence에서 더 큰 상승 폭을 보였다. 이에 더해 단어를 역순으로 배치하는 방식으로도 성능을 향상시켰다.

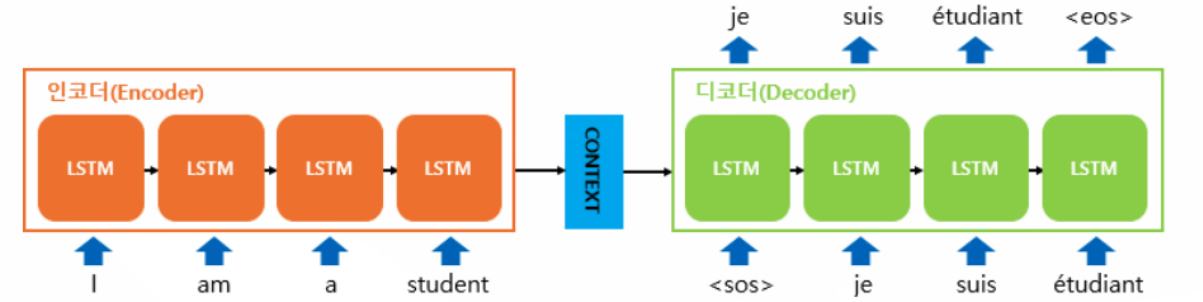
The model

RNN은 기본적으로 sequencial problem에 매우 적절한 model이다. 하지만 input size와 output size가 다른 경우에 대해서는 좋은 성능을 보일 수 없었다. 또한 **장기 의존성 문제가 발생할 수 있다.**

LSTM은 장기적 의존성 문제 또한 학습할 수 있다고 알려져있다. 따라서 LSTM은 이러한 전략을 성공적으로 수행할 수 있을 것이다.



여기서 표시된 LSTM은 "A", "B", "C", "<EOS>"의 표현을 계산한 다음 이 표현을 사용하여 "W", "X", "Y", "Z", "<EOS>"의 확률을 계산.



$$h_t = sigmoid\left(W^{hx}x_t + W^{hh}h_{t-1}\right)$$

$$y_t = W^{yh}h_t$$

$$p\left(y_1, \cdots, y_{T'} | x_1, \cdots, x_T\right) = \prod_{t=1}^{T'} p\left(y_t | v, y_1, \cdots, y_{t-1}\right)$$

본 논문에서 제시하는 model은 Encoder LSTM에서 하나의 context vector를 생성한 뒤 Decoder LSTM에서 context vector를 이용해 output sentence를 생성하는 방식으로 RNN의 한계점을 극복하고자 했다. **input과 output sentence 간의 mapping을 하는 것이 아닌, input sentence를 통해 encoder에서 context vector를 생성하고, 이를 활용해 decoder에서 output sentence를 만들어내는 것이다.**

Encoder LSTM의 output인 context vector는 Encoder의 마지막 layer에서 나온 output이다. 이를 Decoder LSTM의 첫번째 layer의 input으로 넣게 된다. 여기서 주목할만한 점은 input sentence에서의 word order를 reverse해 사용했다는 것이다. 또한 (End of Sentence) token을 각 sentence의 끝에 추가해 variable length sentence를 다뤘다.

우리의 실제 모델은 세 가지 중요한 면

1. **서로 다른 두 가지 LSTM을 사용**

   하나는 **input sequence용이고 다른 하나는 output sequence용**
   그렇게 하는 것이 학습해야 할 **model parameter의 수는 거의 증가시키지 않으면서도 LSTM이 다양한 언어쌍을 동시에 학습하는 것이 가능해지기 때문**

   예를 들자면, English 언어 타입의 Input Sequence를 Representation Vector로 바꾸는 Encoder(LSTM 모델)에 해당 Representation Vector를 특정 언어 타입(French, Korean)의 Output Sequence로 바꾸는 Decoder들을 쌍으로 묶을 수 있다는 말

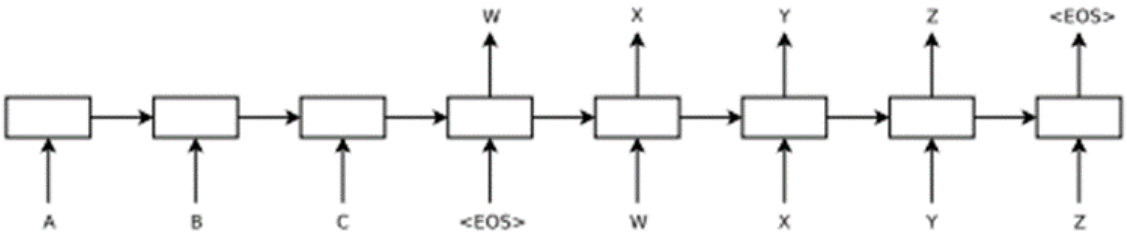
   English->French, English->Korean 이런식으로

2. **깊은 LSTM이 얕은 LSTM보다 성능이 훨씬 뛰어나서 4개의 레이어가 있는 LSTM을 선택**

3. **입력 문장의 단어 순서를 반대로**

   예를 들어 문장 **a, b, c**를 문장 α, β, γ에 매핑하는 대신 LSTM에 **c, b, a**를 α, β, γ로 매핑하도록 요청합니다. 여기서 α, β, γ는 a, b, c의 번역입니다.

   이렇게 a는 α에 아주 가깝고, b는 β에 아주 가까워지므로(Encoder에서는 역순 input sequence를 넣기 때문에 **a가 마지막으로 들어가게 되며** Decoder에서는 **α가 가장 먼저 나오니까** a와 α를 가장 가까이 위치하도록 만든셈.), SGD가 input과 output 사이의 연관 관계를 연결하여 계산하는 것을 쉽게 만들어줍니다. (Gradient의 Backpropagation에 있어 **α의 Gradient를 받아서 가능한 빨리 a에게 전달시킬 수 있다는 의미**인 듯)



Experiments

사용 데이터셋: WMT 2014 English to French

3억 4천 8백만 프랑스 단어/ 3억 4백만 영단어로 구성한 1200만개의 문장 집합들을 학습

source / target language 각각에 fixed size vocabulary를 사용했다 (source: 160,000 / target: 80,000)→ **가장 자주 쓰이는 16만개의 단어**, **target 언어에서 가장 자주 쓰이는 8만개의 단어**

어휘록 외의 모든 단어(OOV)는 특수 토큰인 **“UNK”(Unknown)로 치환**했다.

주요 진행 과정

- B개의 부분 추측을 유지하는 Beam Search를 사용하여 적절한 번역을 찾음

→ Beam Search: 가장 확률이 높은 단어를 추출하는 Greedy Decoding의 단점을 어느정도 극복하기 위한 방법 → 임의의 수 n를 지정한 다음 매 시퀀스마다 누적확률이 높은 상위 n개의 단어만 선택하는 것

- timestep 마다 우리는 beam 안의 부분 추측을 어휘록에 있는 가능한 모든 단어들로 확장

  가장 적합한 추측 B개를 제외하고 나머지를 버림

- 추측 결과에 “<EOS>” 심볼이 나타나면 beam으로부터 제거되고 완성된 추측 집합에 추가
- 기준 시스템에 의해 생성된 1000개 베스트 항목을 재채점 하기위해 LSTM을 사용
- LSTM이 **source sentences를 거꾸로 뒤집어서 학습(target sentences는 뒤집지 않음**).

  → perplexity(혼란도)를 5.8에서 4.7로 감소시킬 수 있었고 번역문의 test BLEU 점수를 향상

  → 이는 앞서 말했던 **단기 의존성**을 도입했기 때문이라고 생각

  - LSTM으로 입력된 source sentence의 초반의 단어들은 timestep이 증가하면서 점차 그 영향이 흐려진다

    따라서 source sentence를 역순으로 입력하면 **source sentence의 초반의 단어의 영향이 희미해지지 않으며 최종 Hidden State에 미치는 영향이 증가할 것**

    그리고 그만큼 역**전파 시, 초반 단어들에 대한 학습 효과를 향상시킨다**

    → 초반 단어에 대해서는 이점이 있을 수 있지만 반대의 경우(후반단어) 오히려 예측의 성능면에서나 신뢰면에서 안좋지 않나라는 의문점이 생김

    어느정도 이유를 추론하자면

    sequencial problem에서는 앞쪽에 위치한 data가 뒤의 모든 data에 영향을 주기 때문에 앞에 위치한 data일 수록 중요도가 더 높다고 할 수 있다. 따라서 reverse를 통해 source sentence에서 앞쪽에 위치한 data(word)들의 target sentence에서의 연관 word와의 거리를 줄이는 것은 더 중요도 높은 data에 대해 더 좋은 성능을 보장하게 되는 효과를 낳는 것으로 추론

- 깊은 LSTM이 얕은 LSTM의 성능을 압도하는 것을 발견했는데 LSTM layer가 추가될 때 마다 perplexity가 거의 10% 씩 감소
  - deep LSTM의 hidden state가 더 크기 때문인 것으로 예상

- output에서 8만개 단어를 대상으로 하는 softmax를 적용

이외 상세 내용

- LSTM parameter들을 -0.08~0.08 사이 값을 갖는 균일분포를 따르는 임의의 값으로 초기화
- learning rate 값을 0.7로 고정해놓고 학습을 진행하다가 5 epoch을 학습한 후 부터는 0.5 epoch 마다 learning rate를 절반으로 줄여가면서 모델이 처음의 5 epoch을 포함하여 총 7.5 epoch을 학습할때까지 학습을 진행
- gradient를 얻기 위해 128 sequence의 배치들을 사용했고 gradient를 batch size로 나눔 (즉, 128로 나눔).
- LSTM에서는 vanishing gradient 문제가 잘 발생하지 않지만 exploding gradient 문제가 발생

  그래서 우리는 gradient의 norm(vector의 크기)이 threshold를 초과할때 그것을 scaling 하는 식으로 norm에 강한 제약을 걸었다

  - 각 training batch에서 g를 gradient를 128로 나눈 값이라고 할 때, $s = \|g\|_2$를 계산한다. 그리고 $s > 5$이면, $g = 5g/s$ 로 설정하여 gradient를 scaling 했다.
- 대부분의 문장은 짧지만 어떤 문장은 길기 때문에 , 랜덤하게 선택된 128의 traing sentence들의 minibatch는 많은 짧은 문장과 적은 수의 긴 문장을 갖게 됨→ minibatch의 연산 대부분이 낭비된다.

  이 문제를 해결하기 위해 **minibatch 내의 모든 sentences들이 거의 비슷한 길이**를 갖도록 했고 **연산 속도가 2배로 향상**

Experimental Results

BLEU 점수를 사용하여 번역 품질을 평가

- BLEU: 번역 품질을 측정하기 위한 정량적 지수로 기계가 번역한 문장과 정답 문장 간의 정확도를 비교하여 측정하는 평가지표. 즉 기계 번역기가 번역한 문장이 사람이 정한 정답 문장과 유사할 수록 더 높은 BLEU 스코어를 기록
- SMT: 통계기반 기계번역_Statistifical Machine Translation

Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

: WMT'14 영어에서 프랑스어로 테스트 세트에 대한 LSTM의 성능.
빔 크기가 2인 5개의 LSTM 앙상블은 빔 크기가 12인 1개(단일) LSTM보다 저렴

Method	test BLEU score (ntst14)
Baseline System [29]	33.30
Cho et al. [5]	34.54
Best WMT'14 result [9]	**37.0**
Rescoring the baseline 1000-best with a single forward LSTM	35.61
Rescoring the baseline 1000-best with a single reversed LSTM	35.85
Rescoring the baseline 1000-best with an ensemble of 5 reversed LSTMs	**36.5**
Oracle Rescoring of the Baseline 1000-best lists	~45

: WMT'14 영어에서 프랑스어로 테스트 세트(ntst14)에서 SMT 시스템과 함께 신경망을 사용하는 방법.

주요 결과:

1. 2014 WMT English to French 번역 작업에 대해 오픈 돼있는 **SMT 기반의 1000개 best 모델을 재채점 하기 위해 LSTM을 사용 → BLEU 점수 36.5점을 얻음**

   → SOTA(State of the Art)에 비해 0.5 낮은 BLEU Score를 달성

   OOV가 여전히 존재함에도 SOTA와 동등한 성능을 달성했다는 것은 충분히 의미있음
2. 위에서 언급했듯이 long Sentence에서도 매우 좋은 성능을 보임

Conclusion

1. 우리는 어휘가 제한적이고 문제 구조에 대한 가정을 거의 하지 않는 대규모 **deep LSTM이 대규모 기계번역 작업에서 어휘가 무제한인 표준 SMT 기반 시스템보다 성능이 우수함**을 보여줌
2. **source sentences의 단어를 역순으로 배치 → 단기 종속성이 학습 문제를 더 쉽게 만들어줌**

   = 때문에 가장 많은 단기 종속성을 갖는 encoding 문제를 찾는 것이 중요하다고 결론을 내린다.
3. **매우 긴 문장을 정확하게 번역하는 LSTM의 능력**

LSTM이 제한된 메모리로 인해 긴 문장에서 실패할 것이라고 확신했지만 **역 데이터셋으로 훈련된 LSTM**은 긴 문장을 번역하는 데 어려움이 거의 없었습니다

4. 마지막으로 단순하고 간단하며 상대적으로 최적화되지 않은 접근 방식이 SMT 시스템보다 성능이 우수하므로 추가 작업으로 번역 정확도가 훨씬 더 높아질 수 있다는 점