Simple NMT using seq2seq

고지훈

1. 개요

seq2seq는 RNN(Recurrent Neural Network) 기반의 모델로, sequence를 입력 받아, 입력에 맞는 sequence를 출력할 수 있는 구조이다. seq2seq 모델과 네이버 영어사전 예문 6500여 개의 데이터를 이용하여, 한글 예문을 입력 받으면 영어 예문을 출력하도록 학습하였다.

2. seq2seq (I. Sutskever et al. , 2014)

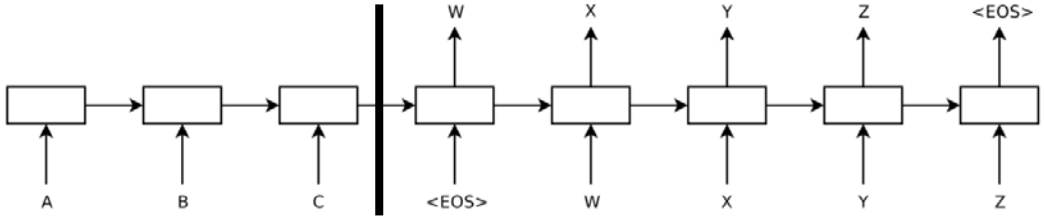


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

seq2seq의 작동원리를 단순화시키면 위 그림과 같이 표현 가능하다. 위의 그림에서 입력받는 sequence는 A B C이고, seq2seq 모델에서 출력하는 sequence는 W X Y Z이다. 알파벳은 임의의 단어를 의미한다.

검은 세로선 왼쪽 부분이 입력받는 sequence의 단어들을 순서대로 배치한 것이고, 오른쪽 부분은 출력하는 sequence의 단어들을 순서대로 배치한 것인데, RNN에서 입력에 따른 출력을 바로 다음 입력에 적용하는 모습을 볼 수 있다. (ex) 그림에서 맨 처음에 <EOS>를 넣었더니 W라는 단어가 나오고, 다시 W라는 단어를 다음 입력에 사용하고 있다.) <EOS>는 실제 sequence의 단어가 아니라, sequence의 시작이나 끝을 알려주는 역할을 한다.

위의 간략화된 그림과 실제 seq2seq model은 아래와 같은 차이가 있다.

- seq2seq를 구현할 때, 하나의 RNN을 사용하는 대신에 검은 세로선 왼쪽 부분 그리고 오른쪽 부분으로 두 개의 RNN으로 나눠서 따로 학습시킨다. (왼쪽 부분을 encoder rnn, 오른쪽 부분을 decoder rnn으로 부른다.)
- vanilla RNN 대신 deep LSTM을 사용하였다. (그러므로 왼쪽 부분의 정확한 명칭은 encoder LSTM, 오른쪽 부분은 decoder LSTM이다.)
- sequence에서 단어들을 입력받을 때 단어 순서를 거꾸로 입력받는다. 즉, 실제 모델에서는 A B C 순서대로 encoder LSTM에 입력하는 대신 C B A 순서로 입력한다.

3. 실제 구현

논문의 내용을 바탕으로 seq2seq 모델을 이용하여 간단한 Neural Machine Translation을 구현하였고, 소스는 https://github.com/jihoon-ko/seq2seq-nmt 를 참고하면 된다.

소스는 Jupyter Notebook 형식이고, seq2seq model을 구현하기 위해 Python 3.6과 TensorFlow 모듈을 사용하였다.

1) 데이터

네이버 영어사전 예문 6540 개를 데이터로 사용하였다.

데이터를 받아오고 파싱하기 위하여 python의 requests와 bs4 module을 사용하였다.

6540개 중에서 1000개의 문장을 validation data로 사용하고, 나머지 5540개 문장을 train data로 이용하였다.

한국어 예문을 입력하면 영어 예문이 출력되도록 seq2seq 모델을 학습시켰다.

2) 상세 구현

한국어 예문의 경우에는 한국어 단어가 지나치게 많다는 문제가 있어서 단어 대신 글자 단위로 쪼개어 사용하였다. 영어 예문의 경우에는 논문과 같이 단어 단위로 쪼개어 사용하였다. 단어와 글자의 개수가 많아서 one-hot 방식을 적용할 수 없다는 문제가 있어서, encoder LSTM에 한국어 글자를 embedding하는 layer를 만들었고 decoder LSTM에 영어 단어를 embedding하는 layer를 추가하였다.

한국어 예문에서 글자 단위로 쪼개어 사용하였더니 성능이 좋지 못하다는 문제가 있어서, word embedding 후 LSTM의 입력으로 넣기 전에 비슷한 위치의 글자들을 효율적으로 처리하기 위하여 convolution 1d layer를 적용시켰다.

또한, 찾아본 다른 구현들에서는 LSTM의 출력 결과에 Dense Layer를 적용한 후 softmax activation을 사용하여 실제 출력 sequence를 얻어냈는데, Dense Layer 대신에 decoder LSTM의 입력 부분에서 사용한 word embedding matrix로 대체하여 출력 sequence를 얻어냈더니 더 좋은 학습 결과를 얻을 수 있었다.

마지막으로, model을 학습할 때 overfitting을 방지하기 위해서 LSTM의 각 셀에 DropoutWrapper를 적용하였다.

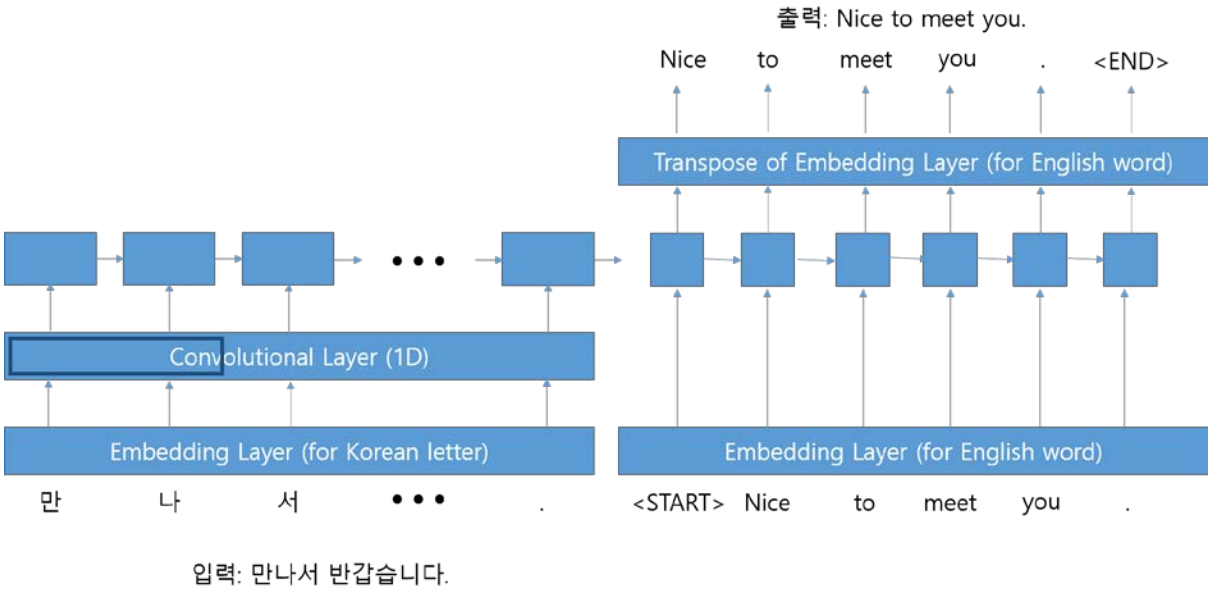
모델을 학습시키는 컴퓨터(CUDA를 지원하지 않음)의 성능 상 문제로 학습 시간을 줄이기 위하여 deep LSTM 대신에 일반적인 LSTM을 사용하였다.

위에 서술한 대로 seq2seq을 커스터마이징하여 만들어진 모델은 다음 페이지의 그림에 나와 있다.

3) 실험 결과

Adam optimizer(Stochastic gradient descent)를 사용하여 학습했으며, epoch는 총 100회로 설정하였다. **그 결과 학습에 사용하지 않은 1000개의 validation sentence 중에 844개의 문장을 올바르게 (실제 영어 예문과 같게) 번역하여 validation accuracy는 84.4%를 기록하였다.**

하지만, 사전에 포함되지 않은 다른 일반적인 문장을 model에 넣어봤을 때에는 좋은 결과를 얻지 못하였는데, 이는 seq2seq를 학습시킬 때 사전의 예문만을 사용하였기 때문이다. 다른 일반적인 문장들이 train 데이터에 포함되어 있었다면, train data에 포함되지 않은 일반적인 문장에 대해서도 비교적으로 좋은 quality의 번역 결과를 얻어낼 수 있을 것이지만, 그러한 한글 - 영어 문장 dataset을 가지고 있지 못해서 부득이하게 크롤링한 사전 예문만을 학습 데이터로 이용하였고, validation data로 사용한 예문에 대해서는 80%가 넘는 비교적 높은 accuracy를 보였다.



4. Reference

Sequence to Sequence Learning with Neural Networks **(I. Sutskever et al. , 2014)**

- https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf