자연어 처리 Assignment2

2019312014 박병준

담당교수: 최윤석

1. Warm up: Read, understand, and reimplement the examples here:

Multi30k_ 데이터셋을 사용하여 독일어를 영어로 번역하기 - nn.Transformer 와 torchtext 사용

마지막 epoch의 loss 값 확인 -> Epoch: 18, Train loss: 0.974, Val loss: 1.929, Epoch time = 44.539s

새로운 문장에 대한 예측 문장 확인 -> A group of people standing in front of an igloo

2. Find the optimal number of epochs and learning rate.

Epoch과 learning rate에 따른 모델 성능을 확인하기 위해, epoch별 train_loss와 val_loss를 비교

	Lr=0.00001		Lr=0.0001		Lr=0.001		Lr=5	
	Train loss	Val loss	Train loss	Val loss	Train loss	Val loss	Train loss	Val loss
Epoch 10	3.819	3.717	1.628	2.008	3.463	3.879	Nan	Nan
Epoch 20	3.158	3.110	0.866	1.942	3.228	3.790	Nan	Nan
Epoch 30	2.735	2.765	0.465	2.084	3.046	3.755	Nan	Nan
Epoch 40	2.425	2.535	0.235	2.315	2.978	3.833	Nan	Nan
Epoch 50	2.181	2.374	0.129	2.517	2.890	3.804	Nan	Nan

Lr=0.00001 경우, 천천히 converge되긴 하지만, epoch이 높아질수록 loss가 꾸준히 낮아지고 있다. 학습이 너무 오래 걸려 더 높은 epoch으론 학습을 진행하지 못했지만, lr=0.00001일 때 optimal한 epoch은 50보다 높은 값이라 볼 수 있다. Lr=0.0001일 때, epoch이 높아질수록 train loss는 0에 수렴할 정도로 낮아지는 반면, val loss는 epoch 20 이후로 오히려 증가하는 경향을 보여 epoch이 30을 넘어가면 overfitting이 일어나는 것을 알 수 있다. Epoch 20 정도에서 early stopping하는 것이 epoch를 많이 설정하지 않아도 모델의 성능을 높일 수 있는 효율적인 방법이라고 생각한다. Lr=0.001이면 training losss는 epoch이 증가할수록 조금씩 줄어들지만, val loss는 줄어들지 않았다. Learning rate가 너무 커 minimum을 찾지 못하는 문제에 직면한 것으로 볼 수 있다. Lr=5일 때는 연산값이 Nan을 기록해 학습이 아예 진행될 수 없음을 알 수 있다.

분석 결과, optimal epoch는 20 내외, learning rate는 0.0001로 설정하는 것이 최적이다.

앞으로의 학습은 epoch=18, lr=0.0001로 통일할 예정이다.

3. Conduct experiments with different number of transformer layer(Encoder/Decoder): 3/3, 3/6, 6/3, 6/6

(epoch=18일 때)	Train loss	Val loss	BLEU score
3/3	0.973	1.927	0.3917070135180366

3/6	1.038	1.922	0.3063552677631378
6/3	1.181	2.017	0.3302822411060333
6/6	1.268	2.006	0.2906414568424225

일반적으로 layer가 더 깊을수록 모델의 성능이 좋아지지만, 여기서는 그렇지 않았다. Learning rate를 0.0001로 정한 것이 영향을 미치지 않았나 생각해본다. 2번 문제에서, epoch이 증가할수록 validation의 loss가 줄어들지 않고 overfitting이 발생했는데, 이 경우도 복잡한 모델로 더 많은 학습이 이루어질수록 loss가 줄어들지 않는 overfitting이 발생한 경우라고 생각한다.

+) learning rate를 0.00001로 재설정하여 6/6 모델을 학습시킨 결과, epoch 18에서 train loss 3.598, val loss 3.594를 기록하여 기대에 못 미치는 성능을 보여주었다. Epoch을 충분히 높게 하면 loss 값이 줄어들 수 있겠지만, 학습 시간이 오래 걸린다는 단점이 있다.

4. Conduct experiments with different number of multi-head: 1, 4, 8

Multi-head	Train loss	Val loss	BLEU score
1	1.026	1.931	0.2798084616661072
4	0.979	1.871	0.42202118039131165
8	0.973	1.927	0.3917070135180366

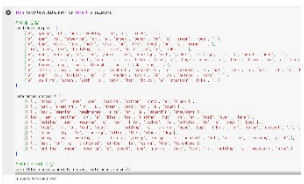
Encoder/Decoder layer 수는 3/3으로 통일했다.

Multi-head 수가 4일 때 가장 낮은 validation loss를 기록하여, 4개의 multi-head가 최적이라는 것을 알 수 있다. Multi-head 수가 8일 때는 4일 때보다 train loss가 줄고 val loss가 늘었다. Head 수가 8인 모델이 1인 모델보다 성능이 좋은데, 다양한 정보를 학습하여 generalization 효과를 얻었기 때문이라고 생각한다.

5. Calculate the BLEU score for the 10 samples. Fill in the last part of the code in the file.

	BLEU score	Predict new sentence
Lr=0.00001(2번 문제의 모델)	0.18327787518501282	"A group of people standing in front of an outdoor event ."
Lr=0.0001	0.49222123622894287	"A group of people standing in front of an igloo"
Lr=0.001	0.07204657202809277	"A group of people are sitting in a circle ."
3/3(3번 문제의 모델)	0.3917070135180366	"A group of people standing in front of an igloo"
3/6	0.3063552677631378	"A group of people stand in front of an igloo ."
6/3	0.3302822411060333	"A group of people stand outside an ATM machines ."
6/6	0.2906414568424225	"A group of people stand in front of an icy monument ."

Epoch=50, lr=0.0001일 때, BLEU score(일치하는 n-gram 개수와 관련)가 가장 높게 나왔다.



코드 사진