Deep Learning: Homework 4

Instructed by *Yi Wu*

Due on May 14, 2021

Runlong Zhou YaoClass 82 2018011309

1.1 Writing Couplets with Sequence-to-Sequence Models

Hyper-parameter tuning

All the combinations of hyper-parameters I've tried for LSTMs (with attention) and Transformers are listed in Table 1.

Model type	Hidden size	#Layers	#Heads	Perplexity
LSTM with attention	1024	1	-	65.7
		2		*62.5*
		3		68.6
	2048	1		63.3
		2		**58.6**
		3		59.5
Transformer	2048	1	32	52
		2	32	48
		4	16	47.5
			32	45.9
			64	47.4
	4096	3	32	45.5
			64	45.0
			128	45.7
		4	64	45.5
		5	64	*44.0*
	8192	6	64	**43.0**

Table 1: Hyper-parameters and corresponding best perplexities on validation set. The boldface perplexities are the best I've got, but the models are too large ($\sim$ 500 MB). Italics are passable perplexities with small model sizes ($\sim$ 250 MB) which are submitted.

Ablation study

Due to resource restraints, vanilla LSTM is only trained on the group of hyper-parameters for submitted LSTM, i.e., hidden size 1024 and 2 layers. The best perplexity on validation set during training is 67.6. The checkpoint file is placed at `generation/models/lstm`$_{s}eq2seq.pt.noattforreference$.

Training curves

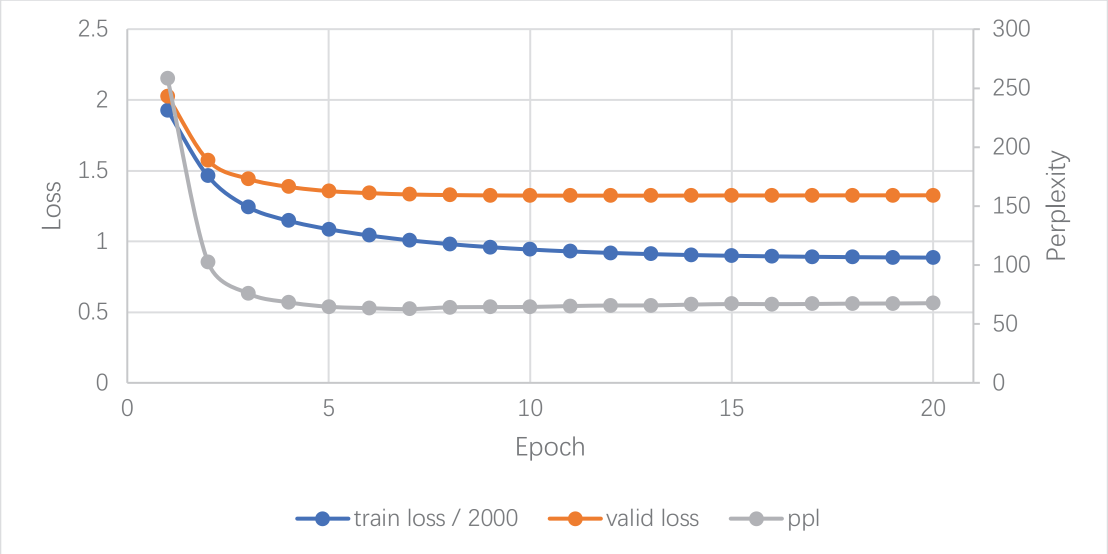


Figure 1: The training curve of the submitted LSTM model.

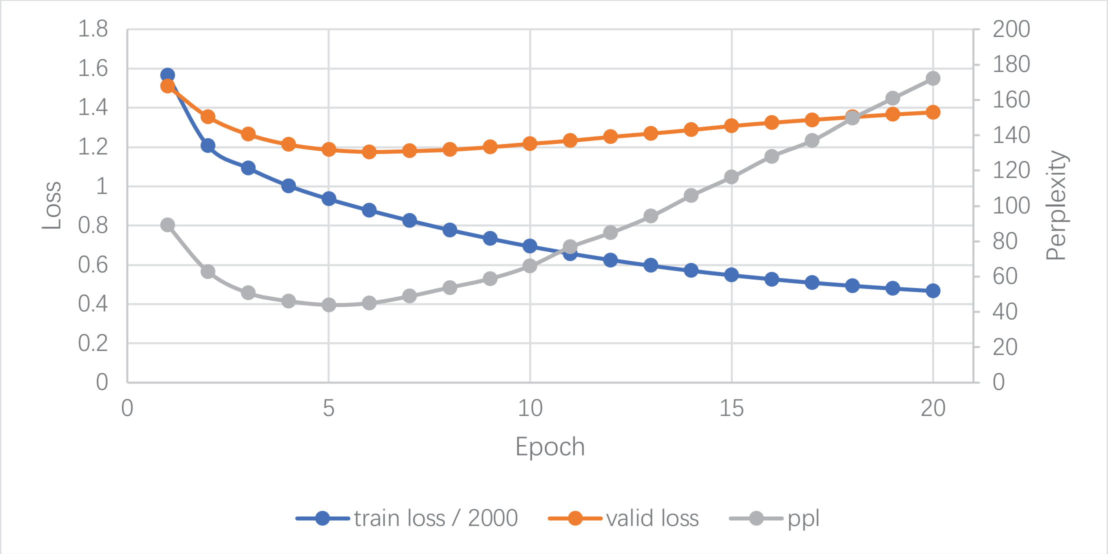
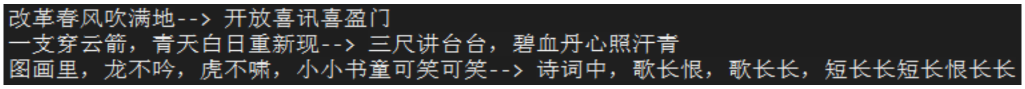


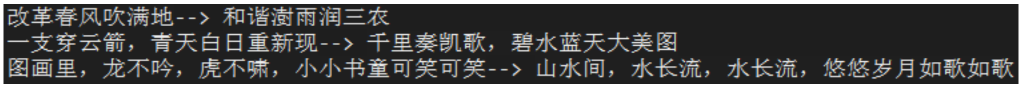
Figure 2: The training curve of the submitted Transformer model.

Generated samples



```
改革春风吹满地--> 开放喜讯喜盈门
一支穿云箭，青天白日重新现--> 三尺讲台台，碧血丹心照汗青
图画里，龙不吟，虎不啸，小小书童可笑可笑--> 诗词中，歌长恨，歌长长，短长长短长恨长长
```

Figure 3: The generated samples of the submitted LSTM model.



```
改革春风吹满地--> 和谐澍雨润三农
一支穿云箭，青天白日重新现--> 千里奏凯歌，碧水蓝天大美图
图画里，龙不吟，虎不啸，小小书童可笑可笑--> 山水间，水长流，水长流，悠悠岁月如歌如歌
```

Figure 4: The generated samples of the submitted Transformer model.

1.2 Writing Poems with Language Models

Hyper-parameter tuning

Due to resource restraints, in this part I only experimented based on the submitted models in part 1.1.

Model type	**Hidden size**	**#Layers**	**#Heads**	**Perplexity**
LSTM	1024	2	-	***113.0***
Transformer	4096	3	64	***89.8***
		4		90.4
		5		90.1
		6		90.7

Table 2: Hyper-parameters and corresponding best perplexities on validation set. The boldface perplexities are the best. Italics are submitted.

Training curves

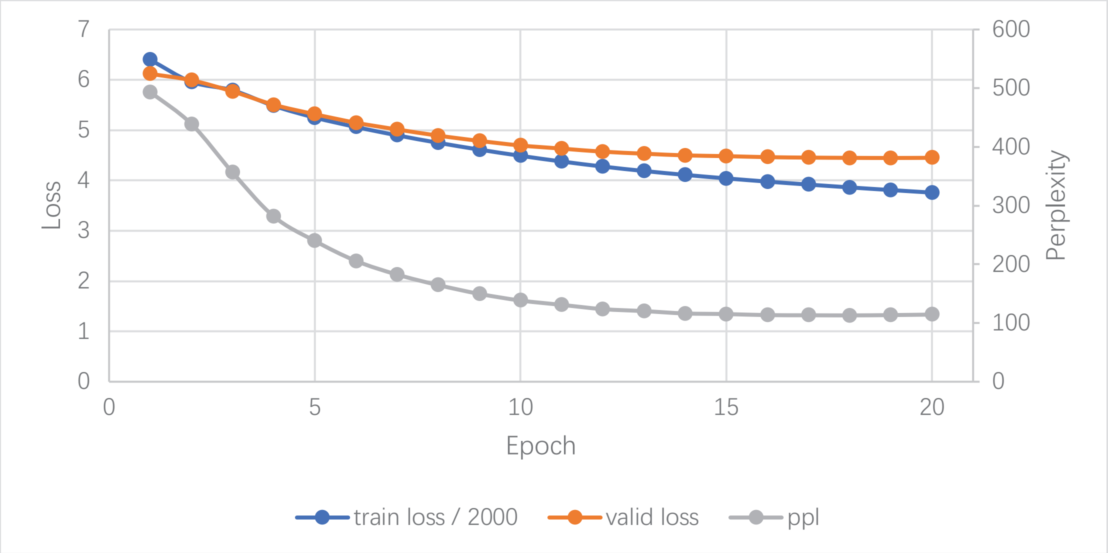


Figure 5: The training curve of the submitted LSTM model.

1.2 Writing Poems with Language Models continued on next page. . .

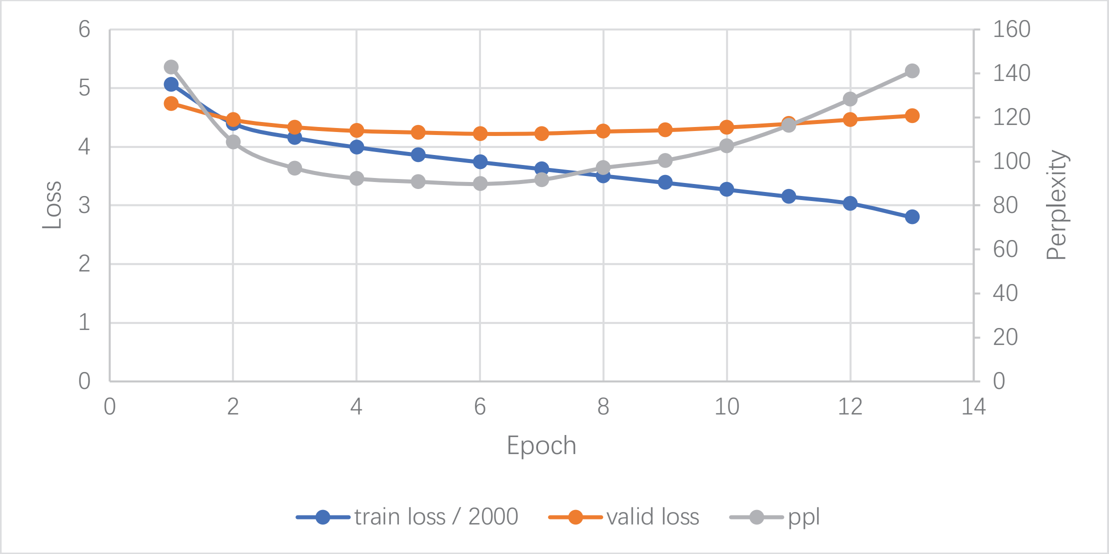
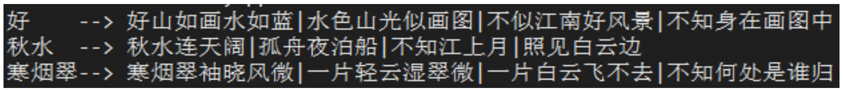


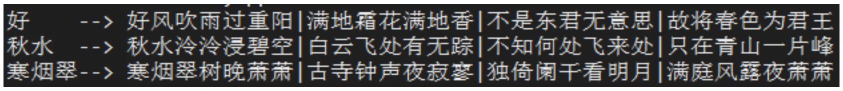
Figure 6: The training curve of the submitted Transformer model.

Generated samples



```
好    --> 好山如画水如蓝|水色山光似画图|不似江南好风景|不知身在画图中
秋水  --> 秋水连天阔|孤舟夜泊船|不知江上月|照见白云边
寒烟翠--> 寒烟翠袖晓风微|一片轻云湿翠微|一片白云飞不去|不知何处是谁归
```

Figure 7: The generated samples of the submitted LSTM model.



```
好    --> 好风吹雨过重阳|满地霜花满地香|不是东君无意思|故将春色为君王
秋水  --> 秋水泠泠浸碧空|白云飞处有无踪|不知何处飞来处|只在青山一片峰
寒烟翠--> 寒烟翠树晚萧萧|古寺钟声夜寂寥|独倚阑干看明月|满庭风露夜萧萧
```

Figure 8: The generated samples of the submitted Transformer model.

2 Classification

Requirements

A custom vocabulary set is used so please copy `classification/vocab.txt` to `Datasets/CLS/vocab.txt`. Also, although the final submitted model do not use pre-trained Bert, please install `transformers`, since when initializing dataset, preprocessed-tokenized inputs are cached for quick retrieve afterwards. Two types of tokenized inputs are preprocessed: LSTM/Transformer style (using `dictionary.py` and `classification/vocab.txt`) for these two types of models, and Bert style (using tokenizer in `https://huggingface.co/bert-base-chinese/tree/main`) for pre-trained Bert model. The Bert tokenized inputs are never fed to models of LSTM or Transformer style.

Implementation

The overall structure comes from Zhang et al. [2019], in which an encoder $e(x)$ and a dual co-matching network $\mathrm{DCN}(P, Q, A)$ are involved. For each question, this model takes P the passage, Q the question and $\mathcal{A} = \{A_i\}$ the set of choices as inputs. There are many options for encoder $e(x)$, I implemented LSTM,

2 Classification continued on next page. . .

Transformer and pre-trained Bert (Bert-base on Chinese corpus). Assume that $e(x)$ maps any input of length L into a hidden feature of size $L \times h$ where h is the hidden size.

First fix a choice A_i, then we calculate a feature matrix $C_i = [M^{pq}; M^{pa_i}; M^{qa_i}]$. Take M^{pq} as example:

$$H^p = e(P),\ H^q = e(Q); \tag{1}$$

$$G^{pq} = \text{RowSoftMax}(H^p W (H^a)^\top),\ G^{qp} = \text{ColumnSoftMax}(H^p W (H^a)^\top); \tag{2}$$

$$E^p = G^{pq} H^q,\ E^q = (G^{qp})^\top H^p; \tag{3}$$

$$S^p = \text{ReLU}(E^p W_1^{pq}),\ S^q = \text{ReLU}(E^q W_2^{pq}); \tag{4}$$

$$M^p = \text{ColumnMax}(S^p),\ M^q = \text{ColumnMax}(S^q); \tag{5}$$

$$g = \text{Sigmoid}(M^p W_3^{pq} + M^q W_4^{pq} + b^{pq}); \tag{6}$$

$$M_{pq} = g M^p + (1-g) M^q. \tag{7}$$

RowSoftMax takes softmax within each row, and ColumnSoftMax is similar; both of them maintains the shape of matrix. ColumnMax takes maximum within each column and returns a row vector. All W_i^{pq}'s and b^{pq} are learnable.

Finally, the probability of choosing A_i is

$$P(A_i|P,Q) = \frac{\exp(V^\top C_i)}{\sum_{j=1}^4 \exp(V^\top C_j)}. \tag{8}$$

V is a vector of dimension $3h$ and is learnable.

Hyper-parameter tuning

Model type	Hidden size	#Layers	#Heads	Batch size	Accuracy (%)
LSTM	512	2	-	4×1	***47.0***
	1024			4×4	45.8
Transformer	512	3	8	1×1	**46.0**
				1×4	45.2
			16	1×1	44.8
				1×4	44.5
Bert	768	-	-	1×4	**41.8**

Table 3: Hyper-parameters and corresponding (current) best accuracies on validation set. The boldface perplexities are the best. Italics are submitted. For batch size of $x \times y$, it means real batch size is x, and accumulating gradients over y batches.

Discussion

When training with pre-trained Bert, the lengths of passages (around 2000) may exceed the limit of Bert (512). My solution to this is to randomly take 4 (maybe overlapping) sections from the passage, each with length 500, and add their hidden features together as a surrogate for the whole passage.

Another problem brought about by the ultra-long passages is high memory consumption. The batch size cannot exceed 4 on a 2080Ti and it drastically affects the training performance.

It can be seen from Table 3 that, Transformers and Bert are beaten by LSTMs. It is convincing that none of the models are adequately trained. The ultimate problem is that all the models are hard to train, especially the pre-trained Bert. Due to resource limitations, it is impossible for me to check every promising

2 Classification continued on next page...

combination of hyper-parameters. One epoch for Bert takes around one hour, so experimenting consumes much time and computing power.

References

Shuailiang Zhang, Hai Zhao, Yuwei Wu, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. Dual co-matching network for multi-choice reading comprehension, 2019.