Deep Learning HW3 Report

唐建宇

2017012221

---

1. 网络结构

RNN

采用四层 LSTM 循环网络，输入词向量和 hidden state 的维度均为 150，从输入到输出每一层加入概率为 0.5 的 dropout 层，将输出的 150 维向量输入全连接层得到最终在词表上的概率分布。

Temporal Attention

- Context Encoder

  采用单层双向 RNN 作为上下文编码器，输入层维度为 150，隐藏层维度为 75（双向 RNN 最终得到输出为 2*nhid=150），在输出层加入概率维 0.5 的 dropout 层。

- 权重计算

  使用两层全连接网络a计算任何一个 context vector h_j 对当前 feature s_i 的权重，输入为h_j, s_i的连接（300 维），输出为一个值即 $e_{ij} = a(s_i, h_j)$，并通过 Softmax 计算归一的权重$\alpha_{ij} = \frac{e_{ij}}{\sum_{k \le i} e_{ik}}$

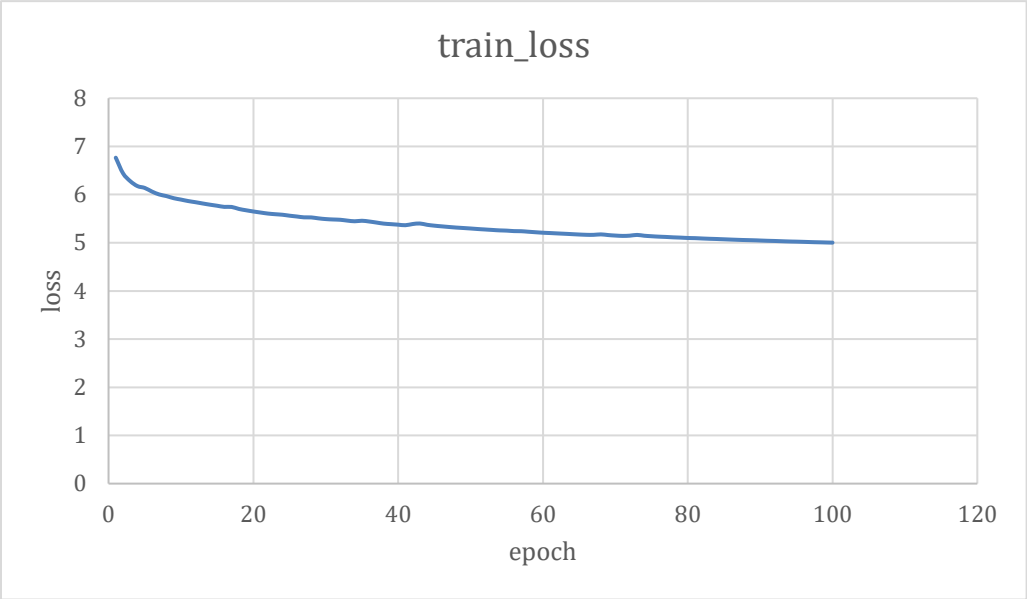
最终得到的 attention 为 context vector 的加权和，即$\sum_{k \le i} \alpha_{ik} h_k$，并将其和 RNN 的输出相加得到最终全连接分类器的输入。

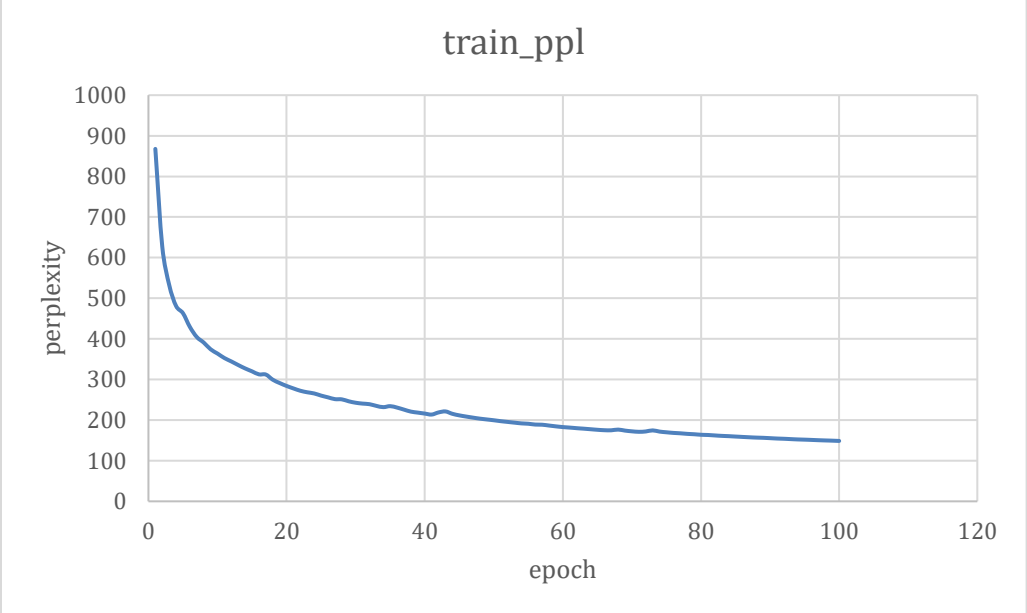
在使用 attention 的过程中，始终保证了$k \le i$，即 Context Encoder 不会看到所预测词及它之后的词，因此避免了标签泄露的可能。

2. 实验结果

最终结果

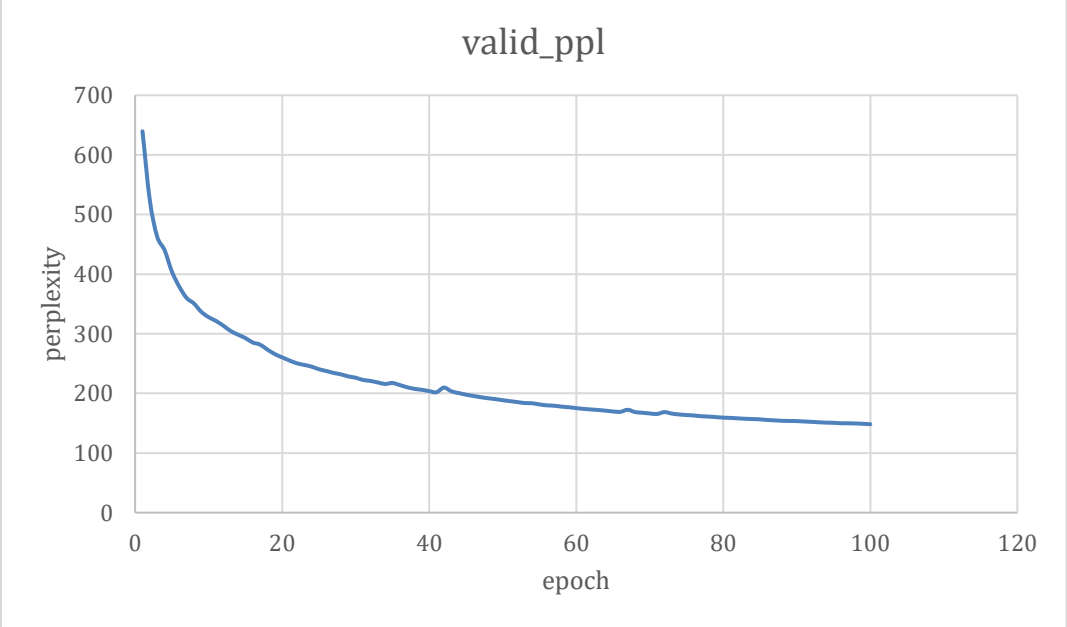
Train



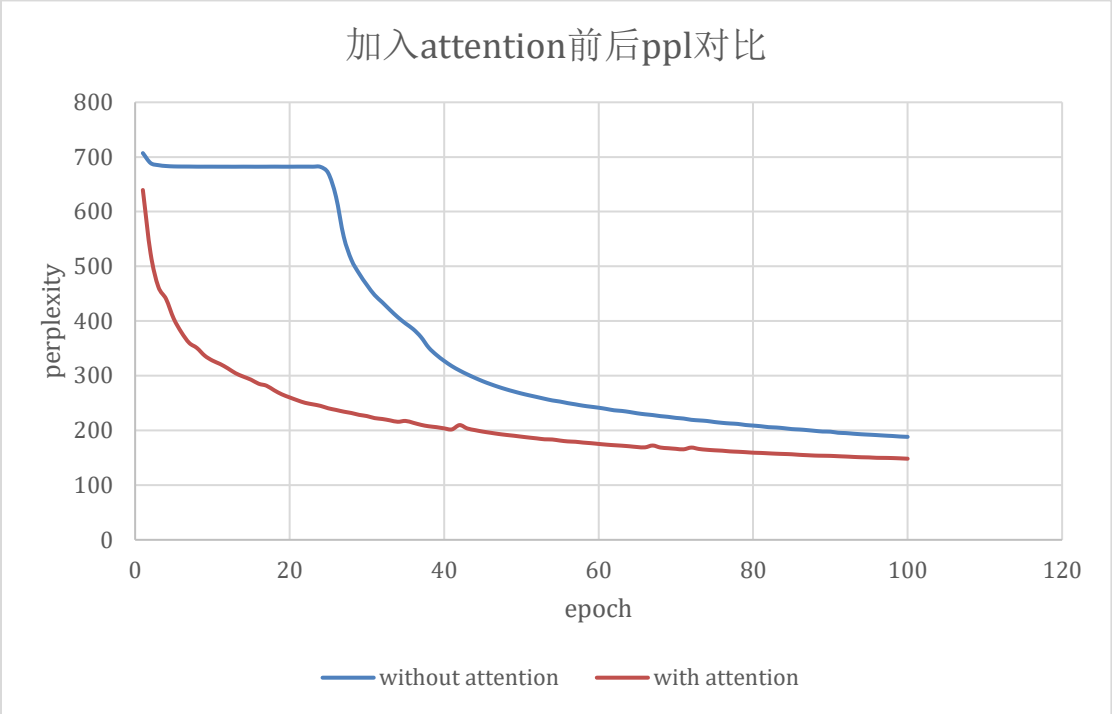


Validation


valid_loss
loss
epoch
0
1
2
3
4
5
6
7
0
20
40
60
80
100
120


valid_ppl
perplexity
epoch
0
100
200
300
400
500
600
700
0
20
40
60
80
100
120

加入 attention 前后的对比

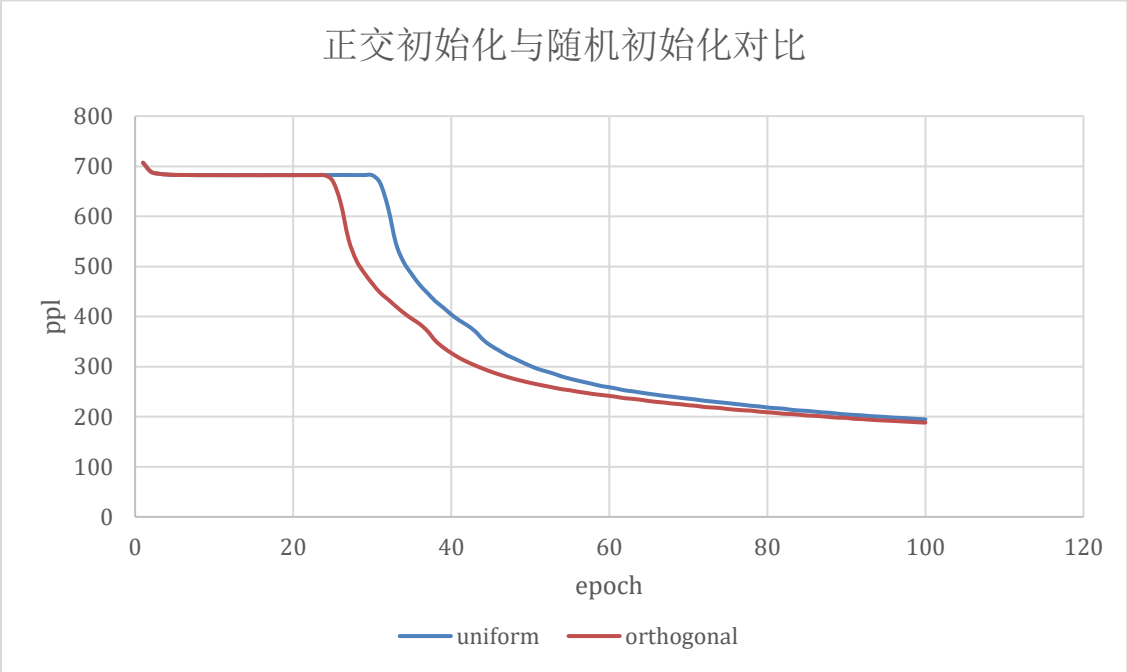


可以看出，加入 attention 后，模型收敛的速度大大加快，直接跳过了刚开始训练时 loss 下降速度很慢的区域，并且最终也取得了更低的 ppl，也证明了 temporal attention 在这个模型上确实起到了很好的效果。

3. 训练技巧

正交初始化

注意到 training curve 在一开始的约三十个 epoch 中，loss 下降的很慢，可能存在模型一开始不容易训练的情况。因此尝试了正交初始化，将参数矩阵初始化为单位正交阵，其特征值的模为 1，这样可以尽量避免在一开始就出现梯度爆炸/消失的现象。分别使用均匀分布初始化和正交初始化的实验结果如下：



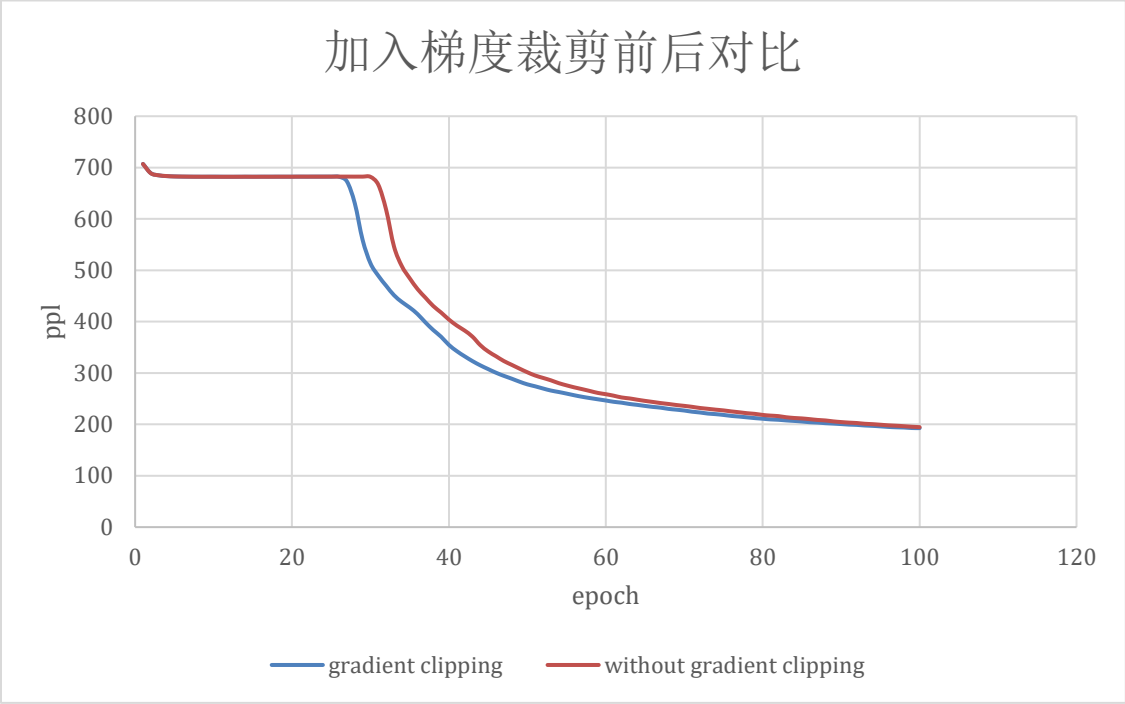
可以看出，采用正交初始化后，相比均匀分布随机初始化的模型，模型收敛的速度有所提升，刚开始训练时 loss 下降很慢的区域有所减小，而最终的结果的差别较小。这也说明通过正交初始化对特征值进行归一取得了有限的效果。

Gradient Clipping

同样针对模型开始收敛速度慢的问题，也尝试通过梯度裁剪的方法进行改善，通过设置梯度的阈值，对梯度进行归一，以避免梯度爆炸或消失。经过实验，尝试了阈值 clip 取 1, 2, 5, 10 四个值时的情况，实验结果如下：

Clip	开始收敛的 epoch
1	32
2	31
5	27
10	27

选取 clip=10 的情况，ppl 的 curve 在加入梯度裁剪前后的对比如下：

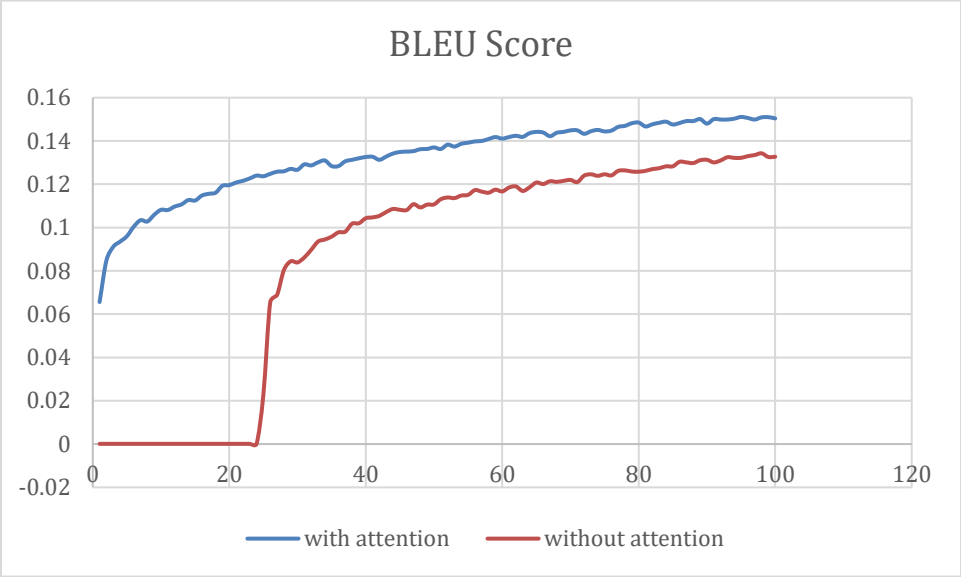


Loss 在第 25 个 epoch 开始快速下降，效果与正交初始化后的效果类似，都是 ppl 下降的时间提前，而最终的结果上看，梯度裁剪的效果与正交初始化类似，只能获得较有限的提升。

4. 评估算法

实现了 BLEU 评估算法，参数 max_n=2，在 1-gram 和 2-gram 的并集上通过 python 的 collections 库的 Counter 类计数，并最终取平均得到 bleu score。实验结果如下：

Result(max_n=2)



在测试集上，加入 attention 的模型达到了 0.1504 的 BLEU 分数；不加 attention 的模型分数为 0.1327。

different max_n

在 BLEU 算法中，max_n 为最大的 n-gram 长度，随着 max_n 增大，要求更长的连续的单词序列翻译正确，因此难度也更大，以 max_n=4 为例，约只有 3%的连续短句能翻译正确。

max_n	BLEU score
2	0.150
3	0.074
4	0.031

5. References

1. Papineni et al. BLEU: a Method for Automatic Evaluation of Machine Translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318.

2. Le, Q., Jaitly, N., & Hinton, G. A Simple Way to Initialize Recurrent Networks of Rectified Linear Units. 2015.

3. Pascanu et al, “On the difficulty of training recurrent neural networks”, ICML 2013.