语言模型任务报告

Student name: *520030910342* 柳纪宇

Course: 智能感知认知实践 – Professor: *Mengyue Wu*, *Xie Chen*

Date: *2023.4.9*

1 概述

在本次实验中，我使用了 RNN、GRU、LSTM、Transformer 等模型完成了 gigaspeech 数据集下的语言模型训练任务，并尝试更改不同的超参数（包括学习率、嵌入层大小、隐层大小、序列长度等）以求在测试集上达到最优预测结果。在模型评测步骤中，我们使用困惑度（Perplexity, PPL）作为模型性能的衡量指标，它的定义如下：

$$PPL = exp(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i|h))$$

此外，我还使用 Tensorboard 记录了模型训练过程中的 train loss、valid loss 和 test loss 的变化情况，log 文件保存在项目根目录下的 runs 目录下。

2 实验过程

2.1 RNN

在这个部分中我使用了 RNN 作为该任务的语言模型进行实验。RNN 是自然语言处理中非常常见且基础的模型，其模型结构如图1(1) 所示。RNN 通过将上一个单元的隐层状态输入下一个单元的方式在全连接神经网络的基础上增加了时序处理能力，并且输入和输出的序列长度能够动态调整，这使得 RNN 在处理序列数据的任务上远远优于传统网络。

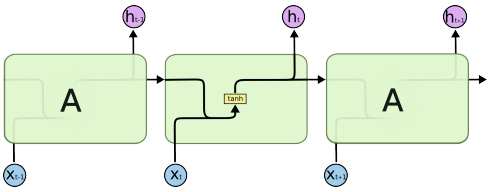

图 1: RNN 模型结构

首先，我分别尝试了使用 relu 和 tanh 作为 RNN 的激活函数进行实验，结果表明使用 relu 作为激活函数的网络极易产生梯度爆炸问题，猜测原因可能是由于 relu 函数

没有对正值进行类似归一化的操作，导致网络的输出在很短的时间内快速增长，同时输出的增长又带来梯度增长的正反馈，故网络无法得到有效的训练。

由于了解到梯度消失和梯度爆炸是 RNN 训练过程中的两大难题，我尝试着在 RNN_relu 模型 baseline 的基础上增大 clip norm 和缩小学习率，得到的结果如下所示：

model	emsize	nhid	nlayers	lr	clip	epoch	batch_size	bptt	dropout	tied	nparameters	test ppl
RNN_TANH	200	200	2	20	0.25	50	20	35	0.2	no	11.75M	756.9
RNN_TANH	500	500	2	20	0.25	50	20	35	0.2	no	28.74M	622.41
RNN_TANH	500	500	2	10	0.25	50	20	35	0.2	no	28.74M	171.13
RNN_TANH	500	500	2	20	0.35	50	20	35	0.2	no	28.74M	748.59

表 1: 不同 clip norm 与学习率条件下 RNN_relu 模型各项指标变化表

结果表明，减小学习率对 RNN 的性能有较大的提升，而增大 clip 值使得 RNN 性能有所下降。

2.2 LSTM

在这个部分中我使用了 LSTM 作为该任务的语言模型进行实验，其模型结构如图2(1) 所示。LSTM 在 RNN 的基础上增添了单元状态 (cell state) 和门 (gate) 的概念。其中单元状态是从先前单元传递过来的信息，而门是 LSTM 控制信息的保留与遗忘比率的工具。在 LSTM 中一共有三个门：控制上个单元隐层状态 h_{t-1} 遗忘比率的遗忘门、控制新信息 x_t 加入比率的输入门，以及控制本单元单元状态 c_t 保留比率的输出门。以上二者的加入使得 LSTM 模型能够在记忆短时信息的同时保留一部分重要的长时信息，从而让模型更好地利用先前输入的历史数据。

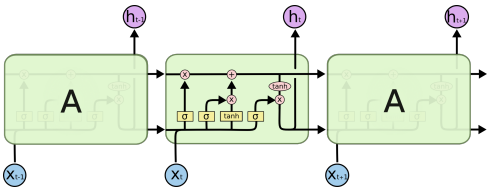

图 2: LSTM 模型结构

首先，我尝试使用不同的 emsize 与 nhid，并尝试将 embedding 层的权重与 softmax 层权重绑定，比较模型最终的参数量与评测结果（其中红色代表不合法的参数量，加粗字体代表最佳结果）：

从中我发现 emsize 和 nhid 在 200 至 700 的范围内越大，模型的性能就越好，更大的 embedding 层与 hidden 层能够帮助模型更好地拟合出满足真实分布的结果。而太大的 emsize 和 nhid 会造成资源浪费与过拟合等问题。

接着，我尝试了在将 emsize 和 nhid 都设置为 500 的前提条件下调节模型训练过程中梯度裁剪的最大范数值，将其设置为 0.15，0.25，0.35，得到的结果如下所示：可以看

model	emsize	nhid	nlayers	lr	clip	epoch	batch_size	bptt	dropout	tied	nparameters	test ppl
LSTM	200	200	2	20	0.25	50	20	35	0.2	no	11.75M	142.2
LSTM	500	500	2	20	0.25	50	20	35	0.2	no	31.75M	125.77
LSTM	500	500	2	20	0.25	50	20	35	0.2	yes	31.75M	125
LSTM	700	700	2	20	0.25	50	20	35	0.2	no	46.67M	123.91
LSTM	700	700	2	20	0.25	50	20	35	0.2	yes	27.28M	120.54
LSTM	1000	1000	2	20	0.25	50	20	35	0.2	no	71.46M	124.78
LSTM	**1000**	**1000**	**2**	**20**	**0.25**	**50**	**20**	**35**	**0.2**	**yes**	**43.75M**	**120.37**

表 2: 不同嵌入层大小、隐层大小及是否进行权重绑定条件下 LSTM 模型的各项指标变化表

model	emsize	nhid	nlayers	lr	clip	epoch	batch_size	bptt	dropout	tied	nparameters	test ppl
LSTM	500	500	2	20	0.15	50	20	35	0.2	no	31.75M	126.12
LSTM	500	500	2	20	0.25	50	20	35	0.2	no	31.75M	125.77
LSTM	500	500	2	20	0.35	50	20	35	0.2	no	31.75M	125.42

表 3: 不同 clip norm 条件下 LSTM 模型各项指标变化表

到在模型参数量相同的情况下，clip 值在 0.15 到 0.35 范围内的变化对模型预测结果没有显著影响。

然后，我还尝试了在将 emsize 和 nhid 都设置为 500 的前提条件下调节训练过程中梯度回传的长度（bptt），将 bptt 设置为 20，35，50，得到结果如下所示：由上述结果

model	emsize	nhid	nlayers	lr	clip	epoch	batch_size	bptt	dropout	tied	nparameters	test ppl
LSTM	500	500	2	20	0.15	50	20	20	0.2	no	31.75M	131.23
LSTM	500	500	2	20	0.25	50	20	35	0.2	no	31.75M	125.77
LSTM	500	500	2	20	0.35	50	20	50	0.2	no	31.75M	125

表 4: 不同 hptt 条件下 LSTM 模型各项指标变化表

可知，较大的 bptt 能够在梯度回传过程中更有效地更新模型参数，从而达到更好的训练效果。

2.3 GRU

在这个部分中我使用 GRU 作为该任务的语言模型进行实验，其模型结构如图3(1)所示。相比 LSTM，GRU 只使用了一个门就完成了对上个单元信息的保留和遗忘，这使得 GRU 在性能上与 LSTM 相当的同时具有更少的参数量和计算复杂度。

在 GRU 模型下，我尝试了调节模型训练过程中的学习率，在将 lr 设置为 25，20，15，10，5 后得到的结果如下：

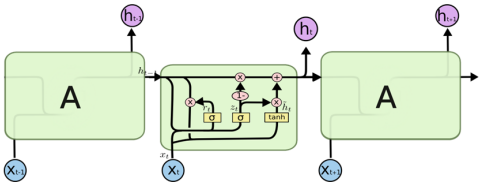

图 3: GRU 模型结构

model	emsize	nhid	nlayers	lr	clip	epoch	batch_size	bptt	dropout	tied	nparameters	test ppl
GRU	500	500	2	20	0.25	50	25	35	0.2	no	30.74M	305.39
GRU	500	500	2	20	0.25	50	20	35	0.2	no	30.74M	205.31
GRU	500	500	2	15	0.25	50	20	35	0.2	no	30.74M	164.92
GRU	500	500	2	10	0.25	50	20	35	0.2	no	30.74M	131.15
GRU	500	500	2	5	0.25	50	20	35	0.2	no	30.74M	133.21

表 5: 不同学习率条件下 GRU 模型各项指标变化表

2.4 Transformer

在这个部分中我使用 Transformer 作为该任务的语言模型进行实验，其模型结构如图4(2)。Transfomer 是近年来 NLP 领域的一个重大突破，不同于传统 CNN 卷积层主导的模型结构，Transformer 引入注意力层和前馈层作为编码器与解码器的主体结构，不仅提升了模型的预测性能，而且提高了模型的并行能力。

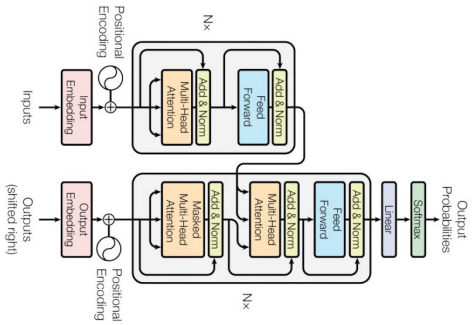

图 4: Transformer 模型结构

首先，我尝试了调节模型训练过程中的学习率，在将 lr 设置为 20，15，10，5 后得到的结果如下：可以看到在 5 至 20 的范围内，学习率降低对模型最终的预测性能有很大的提升效果。过大的学习率很可能导致 Transfomer 模型在学习过程中跳过最优值位置或者导致梯度爆炸。

另外，Transformer 模型还使用了多头注意力机制，通过捕捉不同的注意力信息来提升模型的表达能力。所以我尝试了调节多头注意力层的 head 个数，在将 nhead 设置为 2，4，8 后得到的结果如下：

由该表可知多头注意力层中 head 数量的增加能够带来模型预测性能的提升。更多

model	emsize	nhid	nlayers	lr	clip	epoch	batch_size	bptt	dropout	tied	nhead	nparameters	test ppl
Transformer	500	500	2	20	0.25	50	20	35	0.2	no	4	30.75M	748.37
Transformer	500	500	2	15	0.25	50	20	35	0.2	no	4	30.75M	735.98
Transformer	500	500	2	10	0.25	50	20	35	0.2	no	4	30.75M	182
Transformer	500	500	2	5	0.25	50	20	35	0.2	no	4	30.75M	169.97

表 6: 不同学习率条件下 Transformer 模型各项指标变化表

model	emsize	nhid	nlayers	lr	clip	epoch	batch_size	bptt	dropout	tied	nhead	nparameters	test ppl
Transformer	500	500	2	5	0.25	50	20	35	0.2	no	2	30.75M	173.92
Transformer	500	500	2	5	0.25	50	20	35	0.2	no	4	30.75M	169.97
Transformer	500	500	2	5	0.25	50	20	35	0.2	no	10	30.75M	167.39

表 7: 不同 clip norm 条件下 LSTM 模型各项指标变化表

的 head 能够使 Transformer 模型在计算输入样本的注意力时更加准确，从而使预测结果也更加精确。但当 head 数量达到一定程度后其对模型性能的增强效应也趋于饱和。在模型的实际训练过程中，选取一个适当的 head 个数能够在保证模型性能的时候兼顾计算效率。

3 总结

在本次作业中，我了解了各种语言模型的基本原理与相应特性、使用了 Tensorboard 作为记录模型训练过程的工具，并在各种超参数条件下进行了语言模型训练任务。最终，在测试集上我达到的最佳 PPL 为 120.37，相应的超参数和参数量如下：

model	emsize	nhid	nlayers	lr	clip	epoch	batch_size	bptt	dropout	tied	nparameters	test ppl
LSTM	1000	1000	2	20	0.25	50	20	35	0.2	yes	43.75M	120.37

4 参考文献

[1] https://github.com/roomylee/rnn-text-classification-tf

[2] https://www.analyticsvidhya.com/blog/2019/06/understanding-transformers-nlp-state-of-the-art-models/