神经机器翻译多尺度特征融合及分层优化方法研究报告

摘要：神经机器翻译（Neural machine translation，NMT）是一种利用大型神经网络直接对整个翻译过程进行建模的方法。近年来，各类模型层出不穷，例如基于 RNN、CNN、Attention 的神经机器翻译模型，几乎所有这些神经机器翻译模型都遵循 encoder-decoder 的模型范式。本次课程研究着眼于寻找 encoder 与 decoder 之间、encoder 内部表示的融合方法以及更加高效的优化方法，为深层网络的建模奠定基础。本文针对以下方面进行了实验：（1）encoder 与 decoder 间基于门控机制的信息传递；（2）encoder 端基于 attention 机制的层间特征融合；（3）在模型优化方面使用 layer-wise Adam 优化器。其中，encoder-decoder 间基于门控的特征融合在 iwslt14-de-en 数据集上的 Bleu 达到了 35.75 分

1 研究背景

神经机器翻译模型在多个机器翻译任务上都超越了原有的统计机器翻译模型，并且神经机器翻译端到端的训练模式，大大降低了系统的复杂度。但神经机器翻译模型，尤其是深层神经机器翻译模型，其训练还存在诸如梯度消失、性能退化等较多问题。在这样的背景下，探究神经机器翻译模型如何进行信息的高效传递具有十分重要的意义。

2 研究现状

当前国内外有大量的学者对神经机器翻译模型的结构及信息传递进行研究，比较有代表性的有：DLCL 模型，MSC 模型，transparent transformer 等，本次验将在前人的基础上，继续对 encoder-decoder、encoder 内部的信息传递方式进行实验。

3 研究内容

本次实验主要对以下三个方面进行了探究：基于门控的 encoder-decoder 间信息融合、encoder 层内基于 attention 机制的信息融合、layer-wise Adam 优化器。

3.1 基于门控的 encoder-decoder 间信息融合

3.1.1 模型结构

该模型的出发点受 MSC 框架和 layer-wise attention 模型的启发，其中 MSC 框架使用一个 GRU 网络来对 encoder 的历史信息进行建模，并将其学习得到的历史信息的与来自 encoder 顶层的表示进行融合；在 layer-wise attention 模型中，作者将 encoder 与 decoder 中对应每一层进行了连接，从而促进了 encoder 与 decoder 间的信息流动。

本次实验基于这个路线，从模型可解释性的直观角度进行建模。在 encoder 的顶层，其表示为源语句子整体信息的抽象表示，在 decoder 的底层模型的表示为其学习得到的局部信息，则在 decoder 进行训练的过程中，只使用来自 encoder 顶端的全局抽象表示可能是不够的，所以需要对应 encoder 层的局部信息进行融

合。在本模型中，使用一个共享参数的门控单元将这两个表示进行融合，之所以使用共享参数的门控单元是因为一方面为了控制参数量，另一方面也可以探究 decoder 更加关注哪些信息。其具体结构如图 1 所示。

3.1.2 梯度传递

在引入 encoder-decoder 间的信息融合后，模型的梯度得到了更加高效的传播。在未引入信息融合的标准 Transformer，其梯度传播为：

$$\frac{\partial \mathcal{L}}{\partial layer_e^l} = \frac{\partial \mathcal{L}}{\partial layer_e^L} \times \left(1 + \sum_{j=l}^{L-1} \frac{\partial \mathcal{F}\left(layer_e^j; \Theta_e^{j+1}\right)}{\partial layer_e^l}\right) \tag{1}$$

而引入层间信息融合后，其梯度的反向传播变为：

$$\frac{\partial \mathcal{L}}{\partial layer_e^l} = \frac{\partial \mathcal{L}}{\partial layer_e^L} \times \frac{\partial layer_e^L}{\partial layer_e^l} + \frac{\partial \mathcal{L}}{\partial layer_d^l} \tag{2}$$

由梯度的反向传播公式可以看到，梯度的传播变为多项累加和的形式，这可以有效的减缓梯度消失问题的出现，同时，多条路径的加入，也使得梯度信息的传播更加有效、不失真。

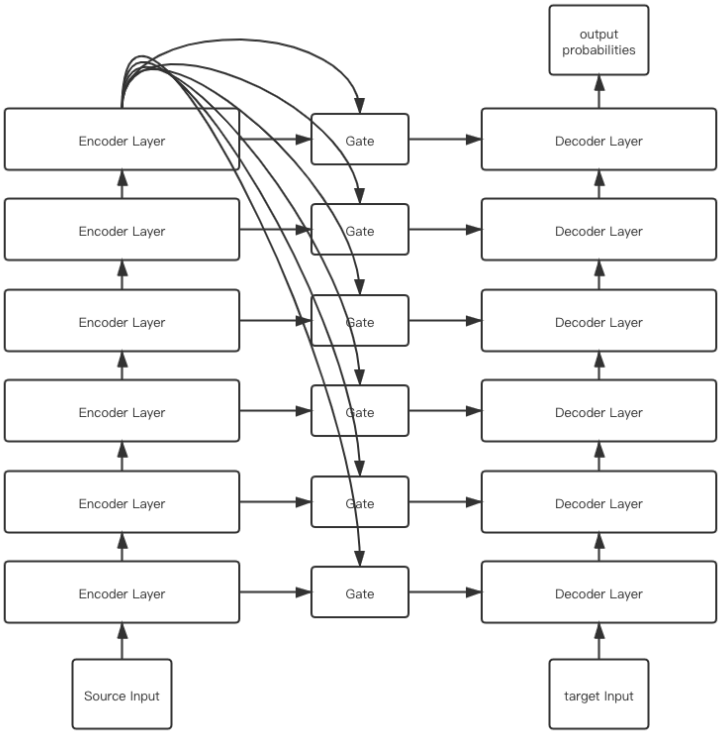


图 1

3.2 layer-wise Adam 优化器

Adam 梯度下降算法是在RMSProp 算法的基础上进行改进的，可以将其看成是带有动量项的RMSProp 算法。该算法在自然语言处理领域非常流行。

Adam 算法的参数更新公式如下：

$$Vdw = \beta_1 Vdw + (1 - \beta_1) dw \tag{3}$$

$$Vdb = \beta_1 Vdb + (1-\beta_1)db \tag{4}$$

$$Sdw = \beta_2 Sdw + (1-\beta_2)dw^2 \tag{5}$$

$$Sdb = \beta_2 Sdb + (1-\beta_2)db^2 \tag{6}$$

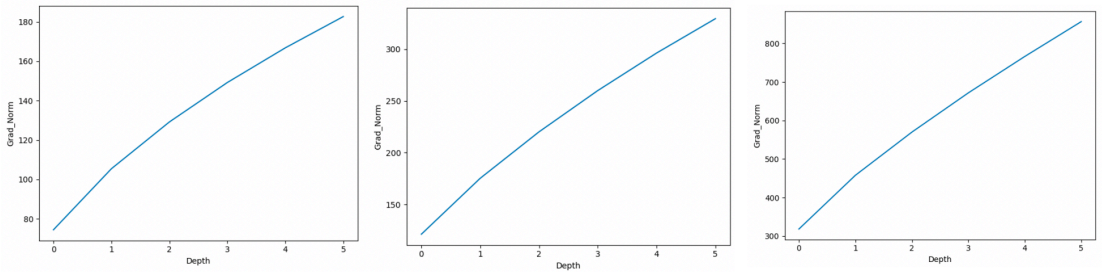
$$V_{dw}^{corrected} = \frac{Vdw}{1-\beta_1^t}, V_{db}^{corrected} = \frac{Vdb}{1-\beta_1^t} \tag{7}$$

$$S_{dw}^{corrected} = \frac{Sdw}{1-\beta_2^t}, S_{db}^{corrected} = \frac{Sdb}{1-\beta_2^t} \tag{8}$$

$$w = w - \alpha \frac{V_{dw}^{corrected}}{\sqrt{S_{dw}^{corrected}}+\epsilon}, b = b - \alpha \frac{V_{db}^{corrected}}{\sqrt{S_{db}^{corrected}}+\epsilon} \tag{9}$$

可以看到Adam 算法相当于在RMSProp 算法中引入了Momentum 算法中的动量项，这样做使得Adam 算法兼具了Momentum 算法和RMSProp 算法的优点：既能使梯度更为“平滑” 地更新，同时可以为神经网络中的每个参数设置不同的学习率。

而对于目前的模型来说，整个模型的不同层使用的是相同的学习率，而影响Transformer训练的一个重要原因是，各层的梯度信息的方差过大，如图2所示。



Var(grad) = 1612.5228　　Var(grad) = 6007.8654　　Var(grad) = 39742.2226

Baseline encoder(1 epochs)　　Baseline encoder(7 epochs)　　Baseline encoder(51 epochs)

图 2

在本次实验中，提出了以下猜想：

（1）高层的梯度范数较大，所以更新步长应较短一些；低层的梯度范数较小，所以更新步长可较长一些。其迭代算法变为：

$$w = w - \alpha \times depth\_scaled \times \frac{V_{dw}^{corrected}}{\sqrt{S_{dw}^{corrected}}+\epsilon}, b = b - \alpha \times depth\_scaled \times \frac{V_{db}^{corrected}}{\sqrt{S_{db}^{corrected}}+\epsilon} \tag{10}$$

其中depth_scaled为与当前层数相关的0到1之间的因子。

（2）对不同层的参数使用不同的Adam优化器，从而针对每一层找到合理的状态。其迭代公式为：

$$w_i = w_i - \alpha \frac{V_{dw_i}^{corrected}}{\sqrt{S_{dw_i}^{corrected}}+\epsilon}, b_i = b_i - \alpha \frac{V_{db_i}^{corrected}}{\sqrt{S_{db_i}^{corrected}}+\epsilon} \tag{11}$$

其中i表示针对第i层参数的Adam优化器。

3.3 encoder 层内基于 attention 机制的特征融合

为了实现深层 Transformer 建模，一个非常有效的手段就是融合不同尺度的特征，常见的特征融合方式有：Residual Network、Dense Network、DLCL 等，而本次实验对基于 attention 机制的层间融合进行了实验。

基于 attention 机制的特征融合即将 encoder 的第 n 层的输入，由原来的第 n-1 层的输出，替换为前 n-1 层的输出状态的融合，而这种融合是基于 attention 机制进行的。具体来说，将第 n-1 层的输出与前面所有层的输出进行 attention 计算并进行加和作为第 n 层的输入，其公式为：

$$\sum_{i=1}^{n-1} Softmax(Q_i K^T) V_i \tag{12}$$

其中Q_i、V_i表示第 i 层的输出状态，K 表示第 n-1 层的输出。其具体结构如图 3 所示。

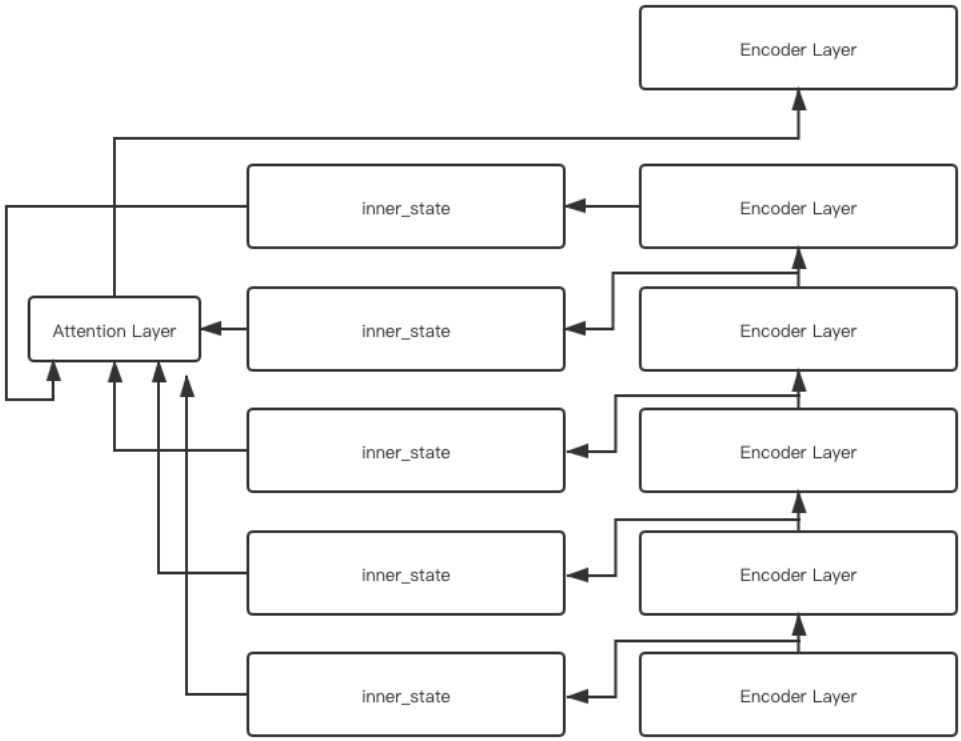


图 3

这样做的直观理解是，在第 n 层，使用第 n-1 层的输出去选择前面所有层的输出，给那些 n-1 层较为“关心”的层的输出给予较大的权重，这样就相当于对前面所有层进行了动态的特征融合。

4 实验

本次实验基于 fairseq 框架，数据集为 iwslt14-de-en，训练使用一个 Nvidia TITAN X GPU。

4.1 基线模型

本次实验所用基线模型为 transformer_t2t_iwslt_de_en，使用 pre-norm 的形式，词嵌入维度为 512 维，多头注意力层的头数为 4 个，模型层数为 6，使用 Adam 优化器（$\beta_1 = 0.9, \beta_2 = 0.98$）。在解码阶段，设置 batch_szie 为 128，束搜索宽度为 4，并平均了最后五个检查点。

4.2 实验结果

encoder 与 decoder 间基于门控单元的融合机制及 encoder 内基于 attention 的特征融合结果如表格 1 所示。

#	Model	De->En
1	baseline	35.81
2	Gate_encoder_decoder	35.75
3	layer attention	33.81

表格 1 其中 gate_encoder_decoder 表示 encoder 与 decoder 间基于门控单元的融合机制，layer attention 表示 encoder 层间基于 attention 的特征融合。

在 layer-wise Adam 优化器中，本实验测试了 1/sqrt(l)、1/l、sqrt(l)三种缩放因子，其中 l 表示当前层数，并且在实验过程中记录了各层的梯度范数的方差变化情况。

#	更新步长	意义	是否收敛	各层梯度方差变化情况	Bleu
1	lr	每一层更新步长为相同学习率	是	1612->39742	35.40
2	lr * (1/sqrt(l))	学习率随层数的增加而缩小	是	1579 -> 14728	30.83
3	lr * (1/l)	学习率随层数的增加而缩小	是	1579->6951	29.70
4	lr * (srqt(l))	学习率随层数的增加而增大	否	1579->209291	
5	lr_i	每一层使用单独的 Adam 优化器进行优化	是		35.04

表格 2 layer-wise Adam 收敛性及模型得分，其中各层梯度变化情况的左端为开始训练时各层梯度二阶范数的方差，右端为训练结束后（51 epochs）各层梯度二阶范数的方差

5 实验分析

5.1 基于门控的 encoder-decoder 间信息融合

由实验结果可知，通过门控单元融合 encoder 与 decoder 的方案是符合直观的，并且不会损伤模型性能，在 iwslt de-en 数据集上虽然相较基线系统下降了 0.06 分，但我认为是由于层正则化和其他编程技巧方面还不够完善所导致。同时，该模型更深远的意义是用于深层网络的建模，通过引入 encoder 与 decoder 的融合机制，使得梯度的传播更加高效。

另一方面，该模型还是基于直观的修改，并没能揭示深刻的数学原理，与底层道理，这也是以后需要改进与考虑的方面。

5.2 encoder 基于 attention 的特征融合

由实验结果可知，该模型对模型的性能有所损伤，该模型的修改较为匆忙所以我认为该损伤来自于编程技巧与正则化等细节的不完善，同时也应深入的分析其底层机制，与数学原理，探索其究竟是否有效。

5.3 layer-wise Adam 实验分析

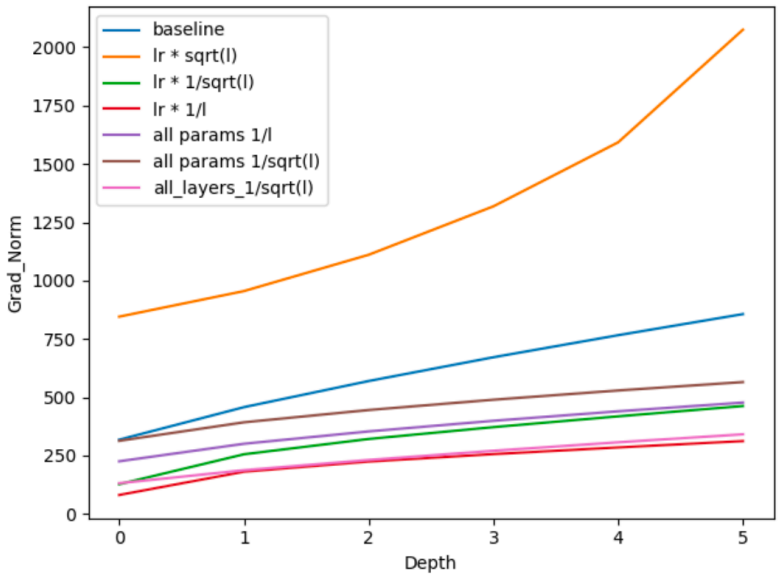


图 4 各种缩放因子的梯度二阶范数随层数的变化趋势（encoder）

由实验结果可以看到，当使用 1/sqrt(l)、1/l 因子时，各层方差明显减小；当使用 sqrt(l) 因子时，模型各层梯度的方差明显增大，并导致了模型无法收敛。这说明，缩小各层梯度间的方差对模型训练的收敛性是有利的，但实验中所使用的缩放因子并未挑选合适，我认为是导致性能下降的主要原因，如图 4 所示，虽然使用 1/l 缩放因子使得方差减小，但其缩放效果过大，却使得整体网络的梯度范数减小，这就相当于整个模型发生了梯度消失。如何选择合适的缩放因子，目前还没有一个有效的方法，这也是以后需要改进的一个方面。

6 总结

首先感谢肖桐老师的悉心教导，通过本次语言分析与机器翻译课程，我充分了解了机器翻译发展的历程，从基于规则的机器翻译到统计机器翻译再到神经机器翻译，也认识到了其背后的数学、语言学、计算机科学、心理学等方面的深层原理。

但本系统也存在着一些问题，如所有模型都未能实现性能的增长，这可能来自于模型细

节的修改、超参数的设置、正则化方式、模型本身的错误等多种因素，本研究的意义在于，通过此次试验构建了完整的翻译系统，并得出了一些有价值的实验现象，为日后的科研工作打下了基础。