文章编号：2095-2163(2024)03-0133-07 中图分类号：TP391 文献标志码：A

并行 RNN 分组策略研究

易也难[1]，卞艺杰[2]
（1 江苏开放大学 商学院，南京 210013；2 河海大学 商学院，南京 211100）

摘 要：并行 RNN 结构或者分组 RNN 结构可以显著减少模型中的参数总量，从而有效地降低模型的训练成本并提高训练效率。本文提出一种高效的并行 RNN 分组策略，该策略不需要对输入数据进行拆分和重组操作，并且可以降低梯度反向传播的不稳定性对于模型训练造成的负面影响。在语言建模和命名实体识别的任务中的实验结果表明，本文所提出的并行 RNN 分组策略，模型的参数计算总量大幅度减少，在 2 个任务中的表现显著提升。

关键词：并行 RNN；分组策略；语言建模；命名实体识别

Research on grouping strategy for parallel recurrent networks

YI Yenan[1], BIAN Yijie[2]
(1 Business School, Jiangsu Open University, Nanjing 210013, China;
2 Business School, Hohai University, Nanjing 211100, China)

Abstract: Parallel RNN structure or grouping RNN structure can significantly reduce the total number of parameters in the model, thus effectively reducing the training cost and improving the training efficiency. This paper proposes an efficient parallel RNN grouping strategy that does not require splitting and reassembling input data, and can reduce the negative impact of gradient back propagation instability on model training. The experimental results of language modeling and named entity recognition show that compared with the traditional method, the total amount of parameter calculation is significantly reduced and the performances in both tasks are significantly improved by using the proposed grouping strategy.

Key words: parallel RNN; grouping strategy; language modeling; named entity recognition

0 引 言

循环神经网络(Recurrent Neural Networks, RNN)具有强大的序列建模能力，近些年被广泛地应用在自然语言处理领域中，并取得了大量优秀的成果，例如语言建模、词性标注、机器翻译、语音识别等等[1-2]。

为了在多种自然语言处理任务中取得更好的结果，研究者们倾向于构建隐层更宽、层数更深的 RNN 模型来处理复杂任务。增加隐层单元的数量，模型能够获得维度更大的参数，从而增强模型的表达能力；增加隐层的层数，不仅模型的隐层单元总数量增多，而且还相应地增加了隐层单元中激活函数的嵌套层数，使模型具备更强大的特征提取能力。从理论上来看，以上这 2 种操作都可以提升模型的性能，使其在复杂任务中取得更好的结果。然而，大尺寸的模型往往会由于过高的参数计算量带来模型训练成本的提升，同时，模型在训练过程中也可能会因为层数过多而产生梯度消失或者梯度爆炸的问题，从而导致训练效率的低下。因此，在实际应用中，通过增加模型的尺寸来提升最终结果，除了需要依靠计算能力更强、内存更高的设备，对模型结构进行改进也是一种有效的途径。

分组的模型结构最早起源于卷积神经网络，分组卷积可以使模型并行计算，提高效率。Krizhevsky 等学者[3]提出 AlexNet 结构，运用了分组的概念将卷积层拆分，在 2 个 GPU 上进行模型的并行化训练。Szegedy 等学者[4]指出单纯地加深和加宽神经网络不仅容易出现过拟合，而且还会占用更多的计算资源，因此在 Inception 结构中引入了稀疏性来提高效率。Zhang 等学者[5]提出 ShuffleNet，也运用了逐点分组卷积方法来减少计算量，以实现移动端的

基金项目：江苏省社会教育规划课题(JSS-L-2023038)。

作者简介：卞艺杰(1964-)，男，博士，教授，主要研究方向：信息管理、电子商务。
通讯作者：易也难(1990-)，男，博士，讲师，主要研究方向：信息管理、电子商务。Email：yi_yenan@163.com

收稿日期：2023-03-27

模型加速。Zhang 等学者[6]提出交错组卷积神经网络(Interleaved Group Convolutional Neural Networks, IGCNets)也使用一种全新的交错组卷积方法,提高了模型整体的计算效率。宋一格等学者[7]在分组卷积的基础上引入了简化的双注意力机制,进一步增强了模型的特征提取能力。赵昊天等学者[8]也运用了分组卷积的方法提高了隐写分析模型的准确率。以上这些研究都是分组结构在卷积神经网络中的应用,取得了较好的结果。

许多研究者们以此为启发,将分组结构应用在 RNN 模型中。Hidasi 等学者[9]最早在推荐系统中应用了并行 RNN 结构,该结构中的每条 RNN 都单独负责某一类数据(图像或文本)特征的提取。Zhu 等学者[10]指出使用并行 RNN 结构可以减少循环连接中一些不相干特征的互相干扰,并且可以在大幅度减少模型中的参数计算总量的同时提高模型的效果。Kuchaiev 等学者[11]将输入数据和长短期记忆模型(Long Short-Term Memory, LSTM)的隐层单元都拆分成若干不相关的分组进行模型的训练,并将这种结构称为 Group LSTM,由于 Group LSTM 的分组之间是相互独立的,该结构非常适用于并行化。Gao 等学者[12]将输入数据拆分后,Group LSTM 无法学习到组与组之间数据特征的相互关系,需要添加一个重组操作去进行补充学习,并且在语言建模、机器翻译等任务中对此方法进行了验证。Yi 等学者[13]利用并行 LSTM 结构生成来自不同子空间的词语表示,用来增强字符级词向量的表达能力,在命名实体识别实验中取得了较好的结果。彭井桐等学者[14]针对门控递归单元(Gated Recurrent Unit, GRU)进行了并行加速优化,对数组做了列维度上的切割,使权重参数能在列维度上并行读取。王茂发等学者[15]也采用了并行 GRU 和双向 RNN 的组合架构,缩短了 RNN 的序列长度,大幅度减少了模型的计算量。

在以上这些研究中,研究者们均在特征维度上对输入数据进行了拆分,再把拆分后的数据分别输入到多组小尺寸的并行 RNN 中进行数据特征的提取。和传统 RNN 相比较,在隐层单元总数量相等的情况下,并行 RNN 结构可以有效降低模型参数的维度大小,从而大幅度减少参数计算总量,提高模型的训练效率。然而,拆分输入数据会不可避免地导致信息的丢失,每组并行 RNN 只能接收到输入数据的一小部分,并且完全独立地进行训练,互相之间没有任何关联,这使得任意 2 组 RNN 接收到的输入数据无法产生联系,模型也就无法学习到之前丢失的信息,需要通过额外的操作来进行弥补。此外,以上研究也没有对并行 RNN 的分组设置进行更深入的讨论,绝大多数都是采用较为简单直接的分组,没有提出如何进行分组才可以使模型获得最佳效果。

针对以上问题,本文提出一种便捷高效的并行 RNN 分组策略,没有将输入数据进行拆分,而是将全部的输入数据都传递给每一组并行 RNN。虽然输入数据的尺寸相比于拆分后的数据有所增大,导致模型的参数计算总量相应增加,但是可以使模型不再需要利用任何的后续操作去学习由于数据拆分而丢失的部分信息,相比于传统 RNN 仍然显著降低了参数计算总量。自由设定模型每一层的分组数量,层与层之间的分组数量不固定。对于多层模型的训练来说,梯度在跨层反向传播时的不稳定性(梯度消失或梯度爆炸)会导致每一层神经元的学习速率不一致,影响最终的训练效果。而通过在每一层设定不同的并行 RNN 分组数量,可以把梯度的传播从原来的单一路径增加至多条路径,每组并行 RNN 参数的梯度大小可以通过分组数量进行控制。当传递到某一层的梯度较大时,在该层设置较多的分组数量,使该层每组并行 RNN 参数的梯度不至于过高。传递到某一层的梯度较小时,在该层设置较少的分组数量,使该层每组并行 RNN 参数的梯度不至于过低。这种自由设定每一层分组数量的策略可以保证模型中所有的神经元按照相对一致的速率进行学习,从而提升模型的训练效果。

本文在语言建模和命名实体识别任务中运用所提出的并行 RNN 分组策略进行了实验。在语言建模实验中利用 Penn Tree Bank(PTB)数据集,采用本文方法构建的并行 LSTM,相比于标准 LSTM 和 Group LSTM 参数计算量更少,效果更好;在命名实体识别实验中利用 CoNLL-2003 英文数据集,本文提出的方法也取得了较好的结果。

1 并行 RNN 的分组策略

1.1 传统 RNN 的弊端

RNN 是一种常用的对序列信息进行建模的神经网络结构,所特有的循环递归结构使其在理论上可以沿着时间维度处理任意长度的序列数据,可以有效地利用之前的信息来计算当前时刻的输出[16]。利用当前时刻的输入结合上一时刻的输出计算得到当前时刻的输出,用式(1)表示:

$$\boldsymbol{h}_l^t = tanh(\boldsymbol{W}[\boldsymbol{h}_l^{t-1}, \boldsymbol{x}_l^t] + \boldsymbol{b}) \tag{1}$$

其中，$\boldsymbol{h}_l^t$ 表示第 l 层的神经元在 t 时刻的输出；$\boldsymbol{x}_l^t$ 表示第 l 层的神经元在 t 时刻接收到的输入；$\boldsymbol{W}$ 和 $\boldsymbol{b}$ 是线性变换时用到的权重和偏置项；$tanh$ 是双曲正切激活函数。

在模型的训练过程中，需要不断计算更新线性变换中用到的参数 $\boldsymbol{W}$ 和 $\boldsymbol{b}$。在只考虑 $\boldsymbol{W}$ 的情况下，假设 $\boldsymbol{h}_l^{t-1}$ 和 $\boldsymbol{h}_l^t$ 都是 m 维向量，$\boldsymbol{x}_l^t$ 是 n 维向量，则 $\boldsymbol{W}$ 的维度可表示为 $[m, m+n]$，那么，该层神经元中需要用式(2)来求得参数总量 RNN_Parm：

$$RNN\_Parm = m*(m+n) = m^2 + mn \tag{2}$$

对于 RNN 的一些变体，例如 LSTM 和 GRU，由于其神经元内部存在多个门结构，控制信息的保存与丢弃，因此参数计算量将会随着门结构数量的增长而倍增[17-18]。

传统 RNN 中需要计算的参数量随着隐层宽度 m 的增加进行二次方增长，如果构建的模型结构中包含较多的隐层单元，那么参数计算量将会急剧增加，这无疑将会大幅度增加训练成本。

此外，传统 RNN 利用梯度的反向传播来进行模型的训练。为了简化问题，这里仅考虑垂直方向上层与层之间的梯度变化情况，省略了水平方向上 t 个时间步的梯度传递，则第 l 层神经元的输出 h_l 可以简化为式(3)：

$$h_l = \tanh(z_l) = \tanh(W_l h_{l-1} + b_l) \tag{3}$$

令代价函数为 C，则第 l 层参数 W_l 上的梯度 δ_l，其定义公式为：

$$\delta_l = \frac{\partial C}{\partial W_l} = \frac{\partial C}{\partial h_l}\cdot\frac{\partial h_l}{\partial z_l}\cdot\frac{\partial z_l}{\partial W_l} = \frac{\partial C}{\partial h_l}\cdot\tanh'(z_l)\cdot h_{l-1} \tag{4}$$

可以继续推导出第 $l-k$ 层参数 W_{l-k} 上的梯度 δ_{l-k}，具体如下：

$$\delta_{l-k} = \frac{\partial C}{\partial h_l}\cdot\tanh'(z_{l-k})\cdot h_{l-k-1}\cdot\prod_{i=0}^{k-1}\tanh'(z_{l-i})\cdot W_{l-i} \tag{5}$$

从式(5)中可以看出，梯度在进行反向传播时，每经过一层都要乘以上一层输出的偏导和参数。当这些数值在 0 和 1 之间时，由于累乘的效果，梯度会越来越小；反之，当这些数值大于 1 时，梯度则越来越大，这就是梯度消失和梯度爆炸的根本原因。这种梯度反向传播的不稳定性会使得多层 RNN 每一层神经元的学习速率不一致，失去了模型深度的意义，可能会导致模型在训练过程中无法有效收敛，从而影响最终的结果。

1.2　多层并行 RNN 结构

针对传统 RNN 的问题，本文现提出一种并行 RNN 的分组策略，将模型结构中的每一层神经元都划分为若干独立的组，每一层中的大尺寸 RNN 将会被划分为若干组并行的小尺寸 RNN，每组并行 RNN 之间相互独立地计算；将这些并行 RNN 的输出进行拼接，作为该层的最终输出传入下一层。设模型层数为 2，每一层神经元数量为 m，现将模型第一层分为 g 组并行 RNN，则每组 RNN 的神经元数量为 m/g，第一层的输出 h_1^t 由该层每组 RNN 的输出 $h1_1^t$，$h2_1^t$，…，hg_1^t 拼接而成，第一层的输出将作为第二层的输入继续进行计算。同样，将第二层分为 k 组并行 RNN，每组 RNN 的隐层单元数量变为 m/k，该模型的最终输出 h_2^t 由第二层每组 RNN 的输出 $h1_2^t$，…，hk_2^t 拼接而成，多层并行 RNN 结构如图 1 所示。

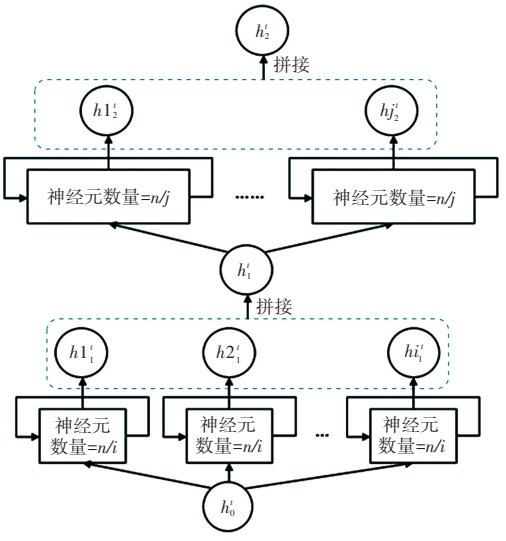


图 1　多层并行 RNN 结构

Fig. 1　The structure of multi-layer parallel RNN

该结构具有以下优势：

(1)可以显著减少模型中的参数计算总量。将分组数量用 g 表示，那么根据参数数量计算公式(2)，将神经元数量 m 替换为 m/g，则参数计算量公式将更新为式(6)：

$$RNN\_Parm\_g = g*\left(\left(\frac{m}{g}+n\right)*\frac{m}{g}\right) = \left(\frac{m}{g}+n\right)*m = \frac{m^2}{g}+mn \tag{6}$$

假设输入的维度和每层神经元的数量相等，即 $m=n$，则可以计算得出在不同的分组数量 g 下，参数计算总量的减少比例，可由式(7)计算求得：

$$Parm\_red = 1 - \frac{\frac{m^2}{g} + m^2}{m^2 + m^2} = \frac{g - 1}{2g} \tag{7}$$

不同分组数量下的参数计算总量减少比例见表 1。

表 1 各分组数量下参数计算总量减少比例

Table 1 Reduction ratio of parameter calculation

分组数量	减少比例/%
2	25.0
3	33.3
4	37.5
5	40.0
6	41.6
7	42.9
8	43.8

与现有的相关研究相比,本文提出的分组策略并没有对输入数据在特征维度上进行拆分,因此在参数计算总量上的减少比例要低;如果对输入数据进行拆分,参数总量会降低得更多。

(2)可以使梯度在反向传播时保持相对稳定。根据 Bengio 等学者[19]研究,梯度爆炸和梯度消失是指模型在训练过程中梯度的范数大幅度增加或趋近于 0,这种梯度的不稳定性会导致模型的学习能力低下。本文使用的并行 RNN 结构每一层的输出是由该层所有并行 RNN 的输出拼接而成,第 l 层的输出 h_l,的定义公式为:

$$h_l = concat(h1_l, h2_l, \cdots, hg_l) \tag{8}$$

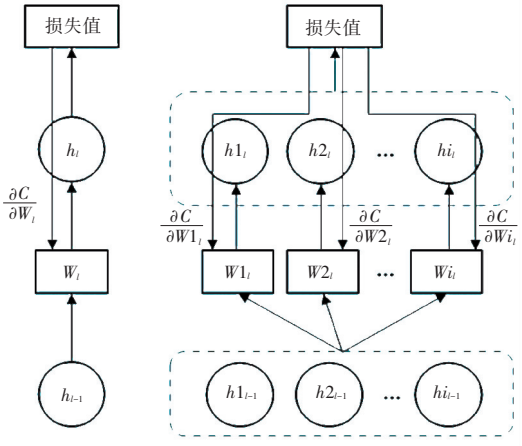
因为该层神经元的总数量不变,所以并行 RNN 输出 h_l 的维度和传统 RNN 相比并没有变化,梯度在反向传播时, h_l 上的梯度也等于每组并行 RNN 上梯度的拼接,可由式(9)表示为:

$$\frac{\partial C}{\partial h_l} = concat\left(\frac{\partial C}{\partial h1_l}, \frac{\partial C}{\partial h2_l}, \cdots, \frac{\partial C}{\partial hg_l}\right) \tag{9}$$

此时,梯度的反向传播由传统 RNN 中的单一路径变为并行 RNN 中分组后的 g 条路径,如图 2 所示。每一条路径传递部分梯度信息至第 i 组并行 RNN 的参数 Wi_l 上,推得的公式为:

$$\delta_{Wi_l} = \frac{\partial C}{\partial Wi_l} = \frac{\partial C}{\partial hi_l} \cdot \frac{\partial hi_l}{\partial zi_l} \cdot \frac{\partial zi_l}{\partial Wi_l} \tag{10}$$

在每层输出维度大小不变的情况下,传递到每一层的梯度大小也相对保持一致,因此,每组并行 RNN 参数上获得的梯度大小将随着该层分组数量的多少而发生变化。该层分组数量越多,则每组并行 RNN 参数获得的梯度就越小;该层分组数量越少,则每组并行 RNN 参数获得的梯度就越大。虽然从模型较高层传递至较低层的总梯度大小仍然会由于跨层传播存在着不稳定的情况,但是由于本文提出的并行 RNN 结构可以自由地设置每一层的分组数量,因此,可以手动地对每组并行 RNN 参数的梯度大小进行控制。



(a) 传统 RNN (b) 并行 RNN

图 2 信息前向传输和梯度反向传播

Fig. 2 Information forward transmission and gradient back propagation

本文提出的并行 RNN 分组策略:首先,在不分组的情况下训练模型一个 *epoch*, 计算每一层并行 RNN 参数的梯度范数,当梯度范数较大时,设置较多的分组数量,梯度范数较小时,设置较少的分组数量;多次尝试不同的设置,以获取最优的分组数量;最后,选择最佳的分组设置对模型重新进行训练,可以保证模型每一层每组并行 RNN 的参数都获得相对一致的梯度大小,从而减少梯度不稳定带来的负面影响,提高模型的训练效率。

2 实验

2.1 语言建模

语言模型是自然语言处理领域中很多重要任务的基础和关键,通过给定的上下文,语言模型可以预测下一个词语出现的概率。通常使用困惑度(*perplexity*)来衡量语言模型效果,困惑度越低,语言模型的效果就越好。Penn Tree Bank(PTB)数据集是目前语言模型训练中使用最广泛的数据集之一,数据质量较高,可以用来评测语言模型的准确性,同时数据量不大,训练速度也比较快。因此,本文将采用 PTB 数据集作为本次的实验数据。

LSTM 模型作为一种带有门控结构的 RNN,可以有效解决长期依赖问题,通常被用来进行语言建

模。因此，本文构建并行 LSTM 模型来进行语言建模的实验，将其和标准 LSTM 模型以及拥有类似并行结构的 Group LSTM 模型进行对比实验。所有的实验都使用一个双层 LSTM 模型，每一层的神经元数量以及词向量维度都设置为 1 500，文本截断的长度为 35，*batch* 的大小为 20，初始化状态为 0，每个 *batch* 计算后得到的最终状态将作为下一个 *batch* 的初始状态，参数的初始化服从 [-0.04, 0.04] 的随机均匀分布，梯度裁剪参数设置为 5.0，在每一层的输出上应用概率为 0.5 的 *dropout*，使用 Stochastic Gradient Descent (SGD) 方法训练所有的模型 55 轮，初始学习率设置为 1.0，并将在第 20 轮后按照每轮 0.8 的比例进行衰减。

对并行 LSTM 模型的分组数量进行最优的设置，以参数的梯度范数大小差值作为衡量分组的标准，差值越小，表明梯度的传播越稳定，该分组数量最佳。通过多次尝试，本文将按照表 2 给出的分组数量进行实验。对于本文所使用的并行 LSTM 模型，用 P-LSTM 表示，括号内的数字从左至右分别表示第一层到最后一层的分组数量。

表 2　双层 LSTM 模型中参数 *W* 的梯度范数大小

Table 2　Gradient norm of parameter *W*

模型	第一层的梯度范数	第二层的梯度范数
标准 LSTM	1.35	1.11
P-LSTM(2-4)	**0.69**	**0.62**

语言建模实验结果见表 3。由表 3 可以看出，与标准 LSTM 模型相比较，本文所使用的并行 LSTM 模型(第一层分 2 组，第二层分 4 组)参数计算总量下降了 31.25%，在 PTB 验证集和测试集上的表现均有所提升，测试集的困惑度值下降了 2.0。Group LSTM 结构参数总量相比于标准 LSTM 下降了 27%，但是并没有提升最终的实验结果，其在 PTB 数据集上的表现基本等价于标准 LSTM。本文所使用的并行 LSTM 结构，在参数计算量最小的情况下，取得了最佳的语言建模实验结果。

表 3　语言建模实验结果

Table 3　Results of language modeling experiment

模型	验证集困惑度	测试集困惑度	参数减少比例/%
标准 LSTM	82.2	78.4	-
2 Group LSTM	82.0	78.6	27.00
P-LSTM(2-4)	**81.1**	**76.4**	**31.25**

2.2　命名实体识别

命名实体识别的主要任务是用正确的实体类型标注文本序列中的每一个词语，实体的类型通常包括人物、地点、组织等等。通常采用 *F*1 得分来衡量命名实体识别的效果，*F*1 得分越高，表示命名实体识别的效果越好。本文利用 CoNLL-2003 共享任务中提供的英文数据集进行实验，该数据集来自路透社语料库，已经预先划分好训练集、验证集和测试集。

本文参考了 Lample 等学者[20]的研究，使用了相同的数据预处理方式以及预训练词向量。在模型结构方面，利用一组 Bi-LSTM 生成字符级的词语表示，将其与预训练词向量拼接后再次输入到另外一组 Bi-LSTM 中对语句建模，并后接一个条件随机场进行最终的命名实体标注。生成字符级词语表示和对语句建模的 Bi-LSTM 隐层单元数量分别设置为 25 和 100，字符向量的初始化服从 [-0.5, 0.5] 的随机均匀分布，梯度裁剪参数设置为 5.0，在每一层的输出上应用概率为 0.5 的 *dropout*，初始学习率设置为 0.1，每一轮训练结束后按照 0.97 的比例进行衰减。训练时监控模型在验证集的表现，并保存在验证集表现最佳的模型进行最终的测试。

针对每一种模型分别进行 5 次试验，并输出验证集和测试集实验结果的均值和标准差，见表 4。本文只进行了分 2 组和分 4 组的并行 Bi-LSTM 模型进行实验，从结果可以看出，当并行 Bi-LSTM 分组数量为 2 时，模型效果最佳，测试集上的 *F*1 得分提升了 0.19，与此同时，参数总量下降了 25%。

表 4　命名实体识别实验结果

Table 4　Results of named entity recognition experiment

模型	验证集 *F*1 得分	测试集 *F*1 得分	参数减少比例/%
BiLSTM	-	90.94	-
P-BiLSTM(2)	**94.08(±0.18)**	**91.13(±0.23)**	**25.0**
P-BiLSTM(4)	93.94(±0.14)	91.08(±0.18)	37.5

将用来对语句建模的 Bi-LSTM 增加至 2 层再次进行实验。由于模型的复杂度提高，本次实验将训练集和验证集数据合并进行训练，之后利用测试集数据进行模型效果的评估，实验结果见表 5。由表 5 可以看出，本文所使用的并行 Bi-LSTM 模型(第一层分 2 组，第二层不分组，P-BiLSTM(2-1))在参数总量减少的情况下，提升了最终的实验结果，相比于传统的双层 Bi-LSTM 模型，*F*1 得分提高了 0.16。

表 5 双层模型的命名实体识别实验结果

Table 5 Results of named entity recognition experiment of two-layer model

模型	第一层参数梯度范数		第二层参数梯度范数		测试集 *F*1 得分
	前向 LSTM	反向 LSTM	前向 LSTM	反向 LSTM	
BiLSTM	1.26	1.14	0.99	0.89	91.46(±0.24)
P-BiLSTM(2-1)	**0.90**	**0.86**	**0.99**	**0.95**	**91.62(±0.14)**

2.3 消融实验

本文提出的并行 RNN 分组策略可以大幅度减少模型的参数计算总量,并且通过在每一层设定不同的分组数量,可以稳定模型各层参数的梯度大小,从而提升模型性能。本文再次利用语言建模任务进行消融实验,以验证所提出的并行 RNN 分组策略是否按照预想的方式对最终结果产生影响。在本次实验中,使用隐层单元数量为 200 的双层 LSTM 模型,在每一层的输出上应用了概率为 0.2 的 *dropout*,初始学习率为 1.0,在第 4 轮后按照 0.5 的比例进行衰减,共训练 13 轮。针对每一种模型(详细模型设置见表 6、表 7)进行 5 次试验,并输出测试集结果的均值和标准差。

首先,对每一种模型设置了不同的分组数量,观察这些不同的分组数量对于模型性能的影响,见表 6。随着每一层分组数量的增多,模型的参数计算总量和训练一个 *epoch* 所需要花费的训练时间逐渐降低。在分组数量为 4 时,模型在测试集上的表现最佳,相比于标准 LSTM,困惑度下降了 2.69,进一步将分组数量提升到 8 组时,模型的效果下降。由此可以看出,在合适的分组数量下,LSTM 模型可以从分组结构中获益,但是当分组数量过多时,单组并行 LSTM 的隐层单元数量过少,不足以从数据中提取到足够的信息,从而对最终结果造成负面影响。

表 6 不同分组数量的模型实验结果

Table 6 Experimental results of different group number

模型	测试集困惑度	参数减少比例/%	训练一个 *epoch* 的时间/s
标准 LSTM	99.77(±0.24)	-	38.30
P-LSTM(2-2)	97.87(±0.32)	25.0	32.84
P-LSTM(4-4)	**97.08(±0.48)**	37.5	27.34
P-LSTM(8-8)	97.23(±0.40)	43.8	26.25

对模型每一层设置不同的分组数量,观察分组数量对于各层参数梯度范数的影响,以及对于模型性能的影响,见表 7。按照每一层分组数量相等、递增和递减这 3 种方式构建模型并与标准 LSTM 进行对比,当分组数量为 2-4 时,模型第一层和第二层参数的梯度范数最为接近,在测试集上的效果也达到了最佳,相比于标准 LSTM,困惑度下降了 2.99。当分组数量为 4-2 时,模型第一层和第二层参数的梯度范数相差最大,在所有模型中也取得了最差的效果。由此可以看出,在并行 LSTM 结构的每一层中设置不同的分组数量,可以控制反向传播至该层参数的梯度大小,选择最佳的分组数量,可以有效地稳定模型整体的梯度反向传播,从而提升模型的最终性能。

表 7 每一层不同分组数量的模型实验结果

Table 7 Experimental results of different group number in each layer

模型	参数梯度范数		测试集困惑度
	第一层	第二层	
标准 LSTM	0.71	0.91	99.77(±0.24)
P-LSTM(2-2)	0.47	0.61	97.87(±0.32)
P-LSTM(2-4)	**0.47**	**0.42**	**96.78(±0.37)**
P-LSTM(4-2)	0.28	0.61	98.10(±0.33)
P-LSTM(4-4)	0.26	0.44	97.08(±0.48)

最后,本文比较了标准 LSTM 和 P-LSTM(2-4)在训练时的收敛速度,如图 3 所示。由图 3 可以看出,由于 P-LSTM(2-4)在训练时具有较为稳定的梯度传播,因此该模型的收敛速度要快于标准 LSTM,训练效果更佳。

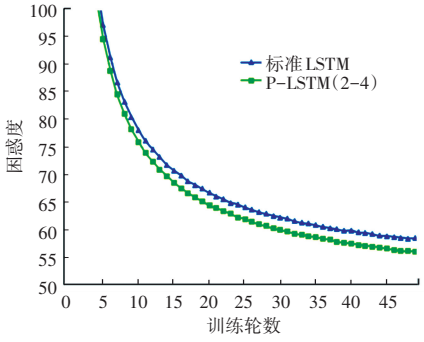


图 3 标准 LSTM 和(2-4) Parallel LSTM 在训练集上的收敛曲线

Fig. 3 Convergence curves of standard LSTM and (2-4) Parallel LSTM on training datasets

3 结束语

本文提出了一种并行 RNN 的分组策略,可以大幅度减少模型的参数计算总量,从而有效降低模型

的训练成本。此外，该策略通过在模型每一层设置不同的分组数量，降低了梯度反向传播时的不稳定性，可以显著提高模型的训练效率，获得更好的效果。在语言建模和命名实体识别任务中应用了本文所提出的分组策略，相比于传统模型均有显著提高，并且参数计算总量也大幅度减少；实证研究也显示了并行 RNN 模型中不同的分组设置对于参数梯度大小以及模型性能的影响，从而证明了本文提出的分组策略是行之有效的。针对并行 RNN 还有很多研究工作，比如：每组并行 RNN 参数的梯度大小虽然很接近，但是仍然存在着些许差异，是否可以进一步缩小这种差异；针对隐层单元的分组是否可以不采用平均分组的方式；以及利用多个 GPU 分别训练每组并行 RNN 是否可以提高训练效率等等。

参考文献

[1] 李启行，廖薇，孟静雯. 基于注意力机制的双通道 DAC-RNN 文本分类模型[J]. 计算机工程与应用，2022，58(16)：157-163.

[2] 易也难，卞艺杰. 基于改进注意力机制的问题生成模型研究[J]. 微电子学与计算机，2022，39(4)：49-57.

[3] KRIZHEVSKY A，SUTSKEVER I，HINTON G E. Imagenet classification with deep convolutional neural networks [J]. Advances in Neural Information Processing Systems，2012，25：1097-1105.

[4] SZEGEDY C，LIU W，JIA Y，et al. Going deeper with convolutions [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston：IEEE，2015：1-9.

[5] ZHANG Xiangyu，ZHOU Xinyu，LIN Mengxiao，et al. Shufflenet：An extremely efficient convolutional neural network for mobile devices[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City，USA：IEEE，2018：6848-6856.

[6] ZHANG Ting，QI Guojun，XIAO Bin，et al. Interleaved group convolutions [C]//Proceedings of the IEEE International Conference on Computer Vision. Venice，Italy：IEEE，2017：4373-4382.

[7] 宋一格，王宁，李宏昌，等. 基于分组卷积与双注意力机制的河流水面污染图像分类[J]. 计算机系统应用，2022，31(9)：250-256.

[8] 赵昊天，钮可，邱枫，等. 基于分组卷积和快照集成的图像隐写分析方法[J]. 计算机应用研究，2023，40(4)：1203-1207.

[9] HIDASI B，QUADRANA M，KARATZOGLOU A，et al. Parallel recurrent neural network architectures for feature-rich session-based recommendations [C]//Proceedings of the 10th ACM Conference on Recommender Systems. Boston：ACM，2016：241-248.

[10] ZHU Danhao，SHEN Si，DAI Xinyu，et al. Going wider：Recurrent neural network with parallel cells[J]. arXiv preprint arXiv：1705.01346，2017.

[11] KUCHAIEV O，GINSBURG B. Factorization tricks for LSTM networks[J]. arXiv preprint arXiv：1703.10722，2017.

[12] GAO Fei，WU Lijun，ZHAO Li，et al. Efficient sequence learning with group recurrent networks[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics：Human Language Technologies，Volume 1 (Long Papers). Louisiana：IEEE，2018：799-808.

[13] YI Yenan，BIAN Yijie. Named entity recognition with gating mechanism and paralell BiLSTM[J]. Journal of Web Engineering，2021，20(4)：1157-1176.

[14] 彭井桐，祝永新，汪辉，等. 基于 FPGA 的 GRU 神经网络飞行数据异常检测[J]. 微电子学与计算机，2021，38(11)：67-73.

[15] 王茂发，冯十辰，黄鸿亮，等. 基于短距空间光谱并行双向 RNN 的高光谱农业图像分类[J]. 南京师范大学学报(工程技术版)，2022，22(4)：1-8.

[16] GOLLER C，KUCHLER A. Learning task-dependent distributed representations by backpropagation through structure [C]//Proceedings of International Conference on Neural Networks (ICNN′96). Washington DC，USA：IEEE，1996，1：347-352.

[17] HOCHREITER S，SCHMIDHUBER J. Long short-term memory [J]. Neural Computation，1997，9(8)：1735-1780.

[18] CHO K，MERRIËNBOER V B，BAHDANAU D，et al. On the properties of neural machine translation：Encoder - decoder approaches[C]//Proceedings of SSST-8，Eighth Workshop on Syntax，Semantics and Structure in Statistical Translation. Doha，Qatar：dblp，2014：103-111.

[19] BENGIO Y，SIMARD P，FRASCONI P. Learning long-term dependencies with gradient descent is difficult [J]. IEEE Transactions on Neural Networks，1994，5(2)：157-166.

[20] LAMPLE G，BALLESTEROS M，SUBRAMANIAN S，et al. Neural architectures for named entity recognition[C]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics：Human Language Technologies. San Diego：ACL，2016：260-270.