基于全局自适应宽度注意力改进的 Transformer

曾庆威 张 建 张鸿昌 谭雨阳 沈文枫
(上海大学 上海 210000)

摘 要 Transformer 在自然语言处理中运用广泛,但存在文本长度过长带来的输入信息被切割、显存占用太大的问题,已有的解决方法是让模型动态决定每层注意力宽度,可以在控制计算量和显存开销的前提下关联最优序列长度,但存在每层最优的注意力宽度并不能达到模型最优注意力宽度的缺点。为此,提出一种全层自适应宽度注意力模型(GAA)。让每层的注意力范围和全局关联,实现模型全局注意力范围最优,还将模型的前馈层修改为带门控单元的前馈层(FFN_{GLU})。在数据集 enwiki8 和 text-8 上的验证表明,该方法仅使用 25% 的训练计算成本,即可达到比基线更好的性能。

关键词 Transformer 全局自适应宽度注意力 FFN_{GLU}

中图分类号 TP3 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.07.022

IMPROVED TRANSFORMER BASED ON GLOBAL ADAPTIVE WIDTH ATTENTION

Zeng Qingwei Zhang Jian Zhang Hongchang Tan Yuyang Shen Wenfeng
(*Shanghai University*, *Shanghai 210000*, *China*)

Abstract Transformer is widely-used in natural language processing, but there is a problem that the input information is cut and the video memory is too large caused by the long text. The existing solution is to let the model dynamically determine the attention width of each layer, and it can associate the optimal sequence length under the premise of controlling calculation amount and memory footprint overhead. However, there is the disadvantage that the optimal attention width of each layer cannot reach the optimal attention width of the model. For this reason, we propose a global adaptive width attention (GAA). We let the attention range of each layer be associated with the global, so as to achieve the optimal global attention range of the model, and modified the feedforward layer of the model to the feedforward layer of the gated unit. Validations on the data sets enwiki8 and text-8 show that this method only uses 25% of the training calculation cost to achieve better performance than the baseline.

Keywords Transformer Global adaptive width attention FFN_{GLU}

0 引 言

自从“Attention Is All You Need”[1]提出以来,Transformer 已成为自然语言处理的主要模型,运用在许多任务上,例如语言建模[2]或句子表示[3]。但其结构需要对输入序列设置一个固定的长度[4](默认长度是 512)。可固定长度下的不同字符所需关联的上下文长度是不同的,有的字符关联的上下文长度远远超过片段长度。简单地将输入文本切分为固定长度不能让模型看到正确的上下文,难以构建更好的表征。同时 Transformer 的注意力机制的计算和显存开销随着输入序列长度增加呈多次方增加,导致其难以处理长文本序列。在面对字符级的任务存在句子长度超过固定长度,无法将句子完整输入模型的问题,只能根据模型固定的输入序列长度将长文本划分为多个片段,导

收稿日期:2021 - 03 - 01。上海智能计算系统工程技术研究中心项目(19DZ2252600);国家重点研发计划项目(2017YFB0701600);上海市科学技术委员会项目(19511121002)。**曾庆威**,硕士生,主研领域:自然语言处理。**张建**,讲师。**张鸿昌**,硕士生。**谭雨阳**,硕士生。**沈文枫**,副研究员。

致分割出来的片段缺失句子的自然边界,在语义上是不完整的。每个片段分开训练带来的片段之间没有联系,使得每个字符之间最长的依赖关系取决于片段的长度。

在控制计算量和显存开销的前提,为了解决固定输入序列长度和长句子切分带来的问题,让模型可以关联最佳的上下文,Transformer XL 设计的缓存机制缓存上一个片段的隐向量,让每个片段可以通过缓存机制关联前一个片段,建立更长的上下文依赖[5]。自适应注意力范围(adaptive-span)Transformer 通过设计掩盖函数让每层的注意力决定需要关注多长的上下文,让模型下层关注较少的上下文,模型上层能够关注更长的上下文,在减少计算量和显存开销下学习到最优的注意力关联[6]。

但 Transformer 的结构是由多个注意力层排列组成[1],不同层之间的注意力范围是相互关联的,每层的注意力范围都是随层数的累加。修改某层的注意力范围会影响其他层的注意力范围,例如下层注意力范围的减少会让上层的注意力范围变相地减少。只是让每层获得最佳的注意力范围并不能让所有层都达到最佳的注意力范围。

为了解决上述问题,本文提出一种全层自适应宽度注意力(Global Adaptive width Attention,GAA)的 Transformer 模型,让模型的注意力层不仅可以调节当前层注意力头的注意力范围,还可以调节模型其他层注意力头的注意力范围,使得层与层之间注意力头的注意力范围保持相互联系,从而保证让每层注意力头学习到最佳的注意力范围的同时,让模型保持最佳的注意力范围。

1 相关工作

在考虑计算效率和显存开销的前提下,通过缓存机制保存前一阶段模型处理好的序列隐向量,关联到当前阶段的序列,在连续的层中重复此机制可以使字符关联更长的距离[7]。

1.1 Transformer XL

Transformer 模型的输入长度必须是固定的,对于大于模型固定长度的序列,需要将输入序列切分一个个固定长度的片段,导致本来应该关联的片段无法关联。为了建立片段之间的依赖性,Transformer XL 设计片段递归机制(Segment-level Recurrence)和相对位置编码机制(Relative Positional Encoding),通过缓存机制保存上一个段的隐藏序列,在对当片段进行处理的时候,缓存并利用上一个片段中所有层的隐向量序列,而且上一个片段的所有隐向量序列只参与前向计算,不再进行反向传播。使模型具有捕获长期依赖的能力,解决了传统 Transformer 固定序列带来的上下文碎片问题,大大提升了其对长序列的处理效果。

1.2 adaptive-span

在保持对显存占用量和计算时间的控制,为了让模型可以关联更长的序列,通过缓存机制保存的模型处理过的序列,再参与注意力计算,可以有效地提高计算效率[8]。但为了模型可以学习到最优的注意力范围,设定的最大缓存长度必须满足最长的注意力范围,但缓存长度的增大同样存在计算和显存开销增大的问题。adaptive-span Transformer 对每层注意力头学到的注意力范围进行分析,发现每层注意力头的注意力范围是不同的,故提出了 adaptive-span 让每层注意力头通过掩盖函数来决定其注意力范围,以此学习到最佳的注意力范围。和传统的 Transformer 的固定注意力范围相比,训练期间根据掩盖函数[9]来决定每层需要关联的序列长度,而不是将保存的序列全部参加计算,可以有效地减少计算开销。

1.3 门控单元

门控单元可以控制信息在网络中流动的路径,使信息可以在模型中畅通无阻地流动。如果没有这些门控单元,信息很可能会在通过多个变换后消失。例如 LSTM 通过由输入门、遗忘门组成的门控单元来实现长期记忆[10]。但 Transformer 不需要遗忘门来解决梯度消失,Dauphin[11]在门控卷积网络中提出只包含输入门的门口单元 GLU,通过 GLU 网络可以决定信息是否能够通过层次结构向前传播,Nal[12]证明了这种机制对于语言建模是有用的,它可以允许模型选择与预测下一个单词相关的单词或特征。

2 方 法

2.1 注意力范围

每个 Transformer 层由一个多头自注意力层(Multi-head Self-attention Layer)[13]和前馈层(FFN)组成,Transformer 模型是由多个 Transformer 层排列组成,每个 Transformer 层的注意力范围是不同的,下层的注意力较小,上层的注意力范围较大,两者之间的注意力范围差距巨大。同时 Transformer 的注意力机制会做多头注意力(Multi-head)处理,每个注意力头的注意力

范围同样不同,有的注意力头只关注附近,有的注意力头的注意力涉及全部上下文。

2.2　注意力范围的关联

在当前层注意力范围不变的情况下,下层注意力头的注意力范围的减少会导致上层注意力头的注意力范围变小,调整模型某层注意力头的注意力范围都会影响其他层的范围,如果每次注意力范围只由当前层决定,并不能让模型的每层学到最好的注意力范围。所以模型学习每层的最优注意力范围,不单单只是调节本层的注意力范围才能达到更好的注意力范围,还可以调节其下层的注意力范围来达到。在一个 12 层的 Transformer 分别加入 GAA 和 adaptive-span,模型学习到每层的注意力范围如图 1 所示,横坐标表示模型的层数,纵坐标表示模型每层学习到的注意力宽度,对比模型添加 GAA 和 adaptive-span 学习到的注意力范围,添加 GAA 的模型所需的注意力范围比 adaptive-span 所需的注意力范围小。添加 GAA 的模型在前三层的注意力范围和 adaptive-span 的注意力范围近似,第 3 层和第 9 层的注意力范围略微增大,其他层的注意力范围都远小于 adaptive-span 的注意力范围,最大减少了 4 288。因为注意力机制的计算和显存开销随着输入序列长度增加成次方增加,所以增大下层注意力减少上层注意力范围可以有效地减少显存开销和训练时间。

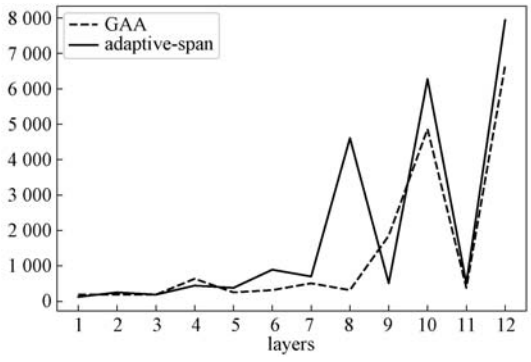

图 1　模型每层注意力范围结构图

相比 adaptive-span,GAA 在学习模型每层注意力范围时,通过调节下层的注意力范围来满足上层的注意力范围需求,缩小了模型的层之间跨度,对整个模型来说在达到最佳注意力范围的同时减少了注意力范围。

2.3　掩盖函数

adaptive-span Transformer 通过添加一个掩盖函数来控制注意力的范围,让模型的每层的注意力头可以动态地控制其注意力的范围。但每层的掩盖函数是孤立的,而每层的注意力范围却是关联的,模型无法整体调节掩盖函数导致每层的注意力范围只受当前层的影响,这样并不能让模型的每层学到最优注意力范围。为了让每层注意力头的注意力范围大小不单单由其当前的层来决定,通过添加参数让每层的注意力范围同时受其他层的影响。这里将 adaptive-span 的掩盖函数修改为:

$$lm_{zl}(x)=\min\left[\max\left[\frac{1}{R}[R+S[z(1-l)+l]-x]\right]\right] \tag{1}$$

式中:z 是每层 head 的参数,l 是所有层共享参数,R 是超参数,设定为 32,S 是最大的注意力范围,设定为 8 192。注意力层处理的计算式变为:

$$self\text{-}attention(x,w_q,w_k,w_v)=\mathrm{softmax}\left(\frac{lm_{zl}(x)(xw_q xw_k)}{\sqrt{d_k}}\right)xw_v \tag{2}$$

式中:$\boldsymbol{W}_q$、$\boldsymbol{W}_k$ 和 $\boldsymbol{W}_v$ 是三个权值不同、尺寸相同的矩阵,输入 x 分别和 $\boldsymbol{W}_q$、$\boldsymbol{W}_k$、$\boldsymbol{W}_v$ 相乘得向量 $\boldsymbol{Q}(xW_q)$、向量 $\boldsymbol{K}(xW_q)$ 和向量 $\boldsymbol{V}(xW_v)$,向量 $\boldsymbol{Q}$、K 和 $lm_{zl}(x)$ 相乘来决定注意力的范围,为了梯度的稳定除以 d[14],再过 softmax 激活函数,最后结果再和向量 $\boldsymbol{V}$ 相乘。

2.4　前馈层

Transformer 的前馈层由两个投射层(Linear)组成,前一个投射层需要过 ReLU 激活函数,其结构如图 2(a)所示,前馈层 FFN 的计算公式如下:

$$FNN=\mathrm{ReLU}(xw_1)w_2 \tag{3}$$

式中:w_1 表示第一个线性变换的参数;w_2 表示第二个线性变换的参数。

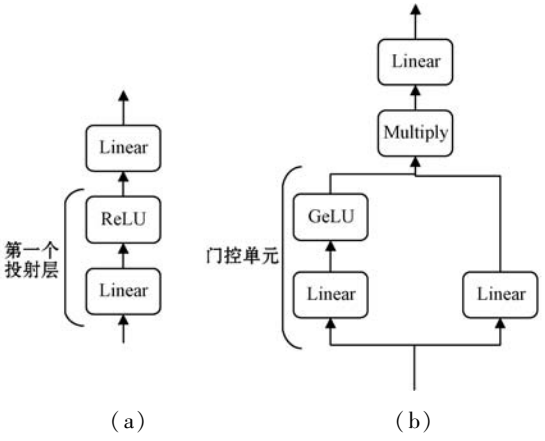

图 2　FFN 的结构图

Transformer 层是注意力层和前馈层组成,在增加模型注意力层的同时也在增加前馈层,但在 Sandwish 中改变注意力层和前馈层排列顺序发现模型的底部包含更多的注意力层,顶部包含更多的前馈层相比固定注意力层和前馈层组成可以提高模型的性能[15]。所

以不改变固定注意力层和前馈层组成,在 FFN 层加入门控线性单元(GLU)来让前馈层有选择地进行线性投射,同样可以达到改变注意力层和前馈层排列顺序的效果。

其结构如图 2(b)所示,将第一个投射层替换为门控单元,让输入经过两个线性变换,其中一个是激活函数设为 GeLU,再将两个线性变换后的分量相乘,其他不变,其计算式为

$$FFN_{\mathrm{GLU}}(x,w_1,w_2,w_3)=(GeLU(xw_1)w_2)w_3 \quad (4)$$

式中:w_1 和 w_2 分别表示 GeLU 第一个线性变换和第二个线性变换的参数。

3 实验与结果分析

在 enwik8 和 text-8 两个字符级数据集(Character level language modeling)上验证本文模型,结果表明 GAA 在满足最优注意力范围的同时减少计算开销,提高了模型训练的速度。

3.1 模型结构

GAA Transformer 模型基于 adaptive-span Transformer,模型的结构如图 3 所示[16],在超参设定上和 adaptive-span Transformer 保持一致,将 GAA Transformer 的层数设为 12,hidden size 设为 512,每层的注意力头设为 8,前馈层的 hidden size 设为 2 048,模型的注意力层和前馈层的 dropout 设为 0.3,模型的最大上下文长度设定为 8 000 K,自注意力层的位置嵌入使用 Shaw[17] 提出的相对位置信息嵌入(Relative Position Embeddings)。

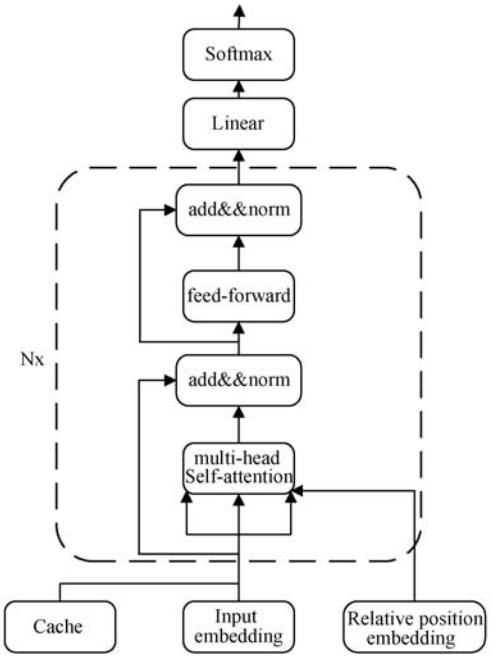

图 3 模型的结构图

但 GAA transformer 在前馈层中加入门控单元,将原先的 ReLU 激活函数换成 GeLU,虽然加入门控单元导致参数量多了 50%,但在达到相同的性能,模型训练速度相比原先的训练速度提高了。在模型的掩盖函数参数设定上,对每层的每个注意力头都设定了参数 z,还设定了其共享的参数 l(z 和 l 都初始化为 0),以此达到全局自适应宽度注意力,让模型学习模型最优注意力跨度。

3.2 实验结果分析

在保证模型的层数、训练超参数等相同的前提,将设计的 GAA Transformer(GAA-T)和 Transformer、Transformer XL(T-XL)、adaptive-span Transformer(adaptive-span)对比。

Enwik8 所有的模型的对比结果如表 1 所示,可以发现 GAA transformer 优于其他基准,在测试集(test)取得最佳结果,达到最低的 PPL 0.99(PPL 是用在自然语言处理领域(NLP)中,衡量语言模型好坏的指标)。由于前两个模型和 GAA Transformer 的性能差距较大,主要和 adaptive-span Transformer 比较。相比 adaptive-span Transformer 需要训练 600 000 K,GAA Transformer 在 test 上在达到更好的性能训练的训练步数降到 200 000 K,减少了三分之一的训练步数。同时可以发现 GAA 的表现和 24 层,277 M 的 T-XL 相同,比 24 层,207 M 的 adaptive-span 仅差 0.01,但所需的注意力范围却减少了 167,减少了模型训练所需的显存。

表 1 不同模型在 Enwiki8 的表现

模型	Layers	Avg Span	Params	Dev	Test
Transformer	12	512	44M	—	1.18
T-XL	12	512	41M	—	1.06
T-XL	24	—	277M	—	0.99
Adaptive-span	12	—	39M	1.04	1.02
Adaptive-span	24	—	209M	1.00	0.98
GAA-T	12	345	51M	1.02	0.99

如表 2 所示,GAA Transformer 在 text-8 数据集的测试集和 adaptive-span Transformer 相比下降了 0.01,平均所需的注意力范围减少了 9,所需的训练步数从 900 000 K 降到 200 000 K,减少了 78% 的训练步数。

表 2 不同模型在 text8 的表现

模型	Layers	Avg Span	Params	Dev	Test
Transformer	12	512	44M	—	1.18
Adaptive-span	12	314	38M	1.05	1.11
GAA-T	12	303	51M	1.04	1.10

总而言之,从模型的平均注意力跨度和训练的步数的减少,表明基于 GAA 相比原先的固定注意力范围和 adaptive-span 更加有效。

4　消融实验

对模型结构加入 GAA 和 FFN_{GLU},为了更好地确定修改的两部分起的作用,本文做了消融实验进行分析,通过分别移除 GAA 设定和 FFN_{GLU}来确定其在模型中起的作用。如表 3 所示,若保留 FFN_{GLU}移除 GAA,模型的训练步数没有变化,但 PPL 上升了,模型的性能变差了,说明添加 GAA 对模型的性能有提升。保留 GAA 移除 FFN_{GLU}模型达到相同的性能训练步数从 150 000 K 增大到 600 000 K,模型的训练时间大幅度增加,这说明 FFN_{GLU}的门控单元可以有效地提高模型的训练速度。

表 3　在 Enwik8 上的消融实验数据表

模型	GAA	FFN_{GLU}	Steps	PPL
GAA-T	有	有	150 000 K	0.99
	有	无	600 000 K	0.99
	无	有	150 000 K	1.02
	无	无	600 000 K	1.02

5　结　语

本文针对 Transformer 自适应宽度注意力存在的每层最优的注意力宽度并不能达到模型最优注意力宽度,引入全层的自适应宽度注意力,让模型调整每层的注意力宽度的时候同时调整全局注意力,让模型学习到最优的注意力范围。在 FFN 使用 GLU 让模型的训练步数缩小了四分之三,更快地收敛到最佳状态。与传统的 Transformer 相比,该改进在增大模型的最大可见上下文长度的同时极大地节约了计算和显存开销。

参　考　文　献

[1] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//31st International Conference on Neural Information,2017:5998 – 6008.

[2] Al-Rfou R, Choe D, Constant N, et al. Character-level language modeling with deeper self-attention[C]//33rd AAAI Conference on Artificial Intelligence,2019:3159 – 3166.

[3] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding [C]//Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,2019:4171 – 4186.

[4] Dai Z H, Yang Z L, Yang Y M, et al. Transformer-XL: Attentive language models beyond a fixed-length context[EB]. arXiv:1901.02860,2019.

[5] Song K, Tan X, Qin T, et al. Mpnet: Masked and permuted pre-training for language understanding[EB]. arXiv preprint arXiv:2004.09297, 2020.

[6] Sukhbaatar S, Grave E, Bojanowski P, et al. Adaptive attention span in transformers[C]//57th Annual Meeting of the Association for Computational Linguistics.

[7] Sukhbaatar S, Szlam A, Weston J, et al. End-to-end memory networks[C]//28th International Conference on Neural Information Processing Systems,2015:2440 – 2448.

[8] Grave E, Joulin A, Usunier N. Improving neural language models with a continuous cache[EB/OL]. [2021 – 03 – 21]. https://ar5iv.labs.arxiv.org/html/1612.04426?fallback=original.

[9] Yacine J, Edouard G, Armand J, et al. Variable computation in recurrent neural networks[C]//5th International Conference on Learning Representations,2017:1880 – 1892.

[10] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation,1997,9(8):1735 – 1780.

[11] Dauphin Y N, Fan A, Auli M, et al. Language modeling with gated convolutional networks[C]//34th International Conference on Machine Learning,2017:933 – 941.

[12] Kalchbrenner N, Espeholt L, Simonyan K, et al. Neural machine translation in linear time[EB]. arXiv:1610.10099, 2016.

[13] Al-Rfou R, Choe D, Constant N, et al. Character-level language modeling with deeper self-attention[C]//33rd AAAI Conference on Artificial Intelligence,2019:3159 – 3166.

[14] Tan M X, Le Q. EfficientNet: Rethinking model scaling for convolutional neural networks[EB]. arXiv:1905.11946v1, 2019.

[15] Press O, Smith N A, Levy O. Improving transformer models by reordering their sublayers[C]//58th Annual Meeting of the Association for Computational Linguistics,2020:2996 – 3005.

[16] Hendrycks D, Gimpel K. Bridging nonlinearities and stochastic regularizers with gaussian error linear units[EB]. arXiv:1606.08415,2016.

[17] Shaw P, Uszkoreit J, Vaswani A. Self-attention with relative position representations[C]//Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,2018:464 – 468.