情报学报 2018年2月 第37卷 第2期
Journal of the China Society for Scientific and Technical Information, Feb. 2018, 37(2): 194-205

DOI: 10.3772/j.issn.1000-0135.2018.02.008

结合 LSTM 和 CNN 混合架构的深度神经网络语言模型

王 毅，谢 娟，成 颖

（南京大学信息管理学院，南京 210023）

摘 要 语言模型是自然语言处理研究中的基础性工作，是计算机识别与理解自然语言的桥梁，是人工智能学科的前沿及热点课题。其在语音识别、机器翻译、信息检索和知识图谱等领域都有着广泛的应用。至今，语言模型已经历了从统计模型、神经网络模型到深度神经网络模型的衍化。随着深度学习技术的广泛应用，采用大规模的数据集、复杂的模型以及高昂的训练代价成为语言模型建模的特点。本文通过模型输入拟人化、卷积神经网络（convolutional neural network）编码以及融合门机制并结合长短时记忆单元（long short-term memory，LSTM）优化了语言模型，提出了结合 LSTM 和 CNN 混合架构的深度神经网络语言模型（Gated CLSTM）。利用深度学习框架 Tensorflow 实现了 Gated CLSTM。实验环节还采用了负采样及循环投影层等经典的优化技术，在包含近十亿个英文单词的通用数据集（one billion word benchmark）下测试了模型的性能，分别训练了单层模型和三层模型，以观察网络深度对性能的影响。结果显示，在四个 GPU 的单机环境下，单层模型经过 4 天的训练，将模型混淆度（perplexity）降低至 42.1；三层模型经过 6 天的训练后将混淆度降低至 33.1；与多个典型的基准模型相比，综合硬件、时间复杂度以及混淆度三个指标，Gated CLSTM 获得了明显的改进。

关键词 语言模型；循环神经网络；卷积神经网络；字符序列编码

Deep Neural Networks Language Model Based on CNN and LSTM Hybrid Architecture

Wang Yi, Xie Juan and Cheng Ying

(*School of Information Management, Nanjing University, Nanjing* 210023)

Abstract: The language model is one of the most important domains in natural language processing. It is a bridge for the computer to identify and comprehend human language, and it is also a sign of Artificial Intelligence development. The language model is popular in Speech Recognition, Machine Translation, Information Retrieval, and Knowledge Mapping. With the rapid expansion of technology and hardware, the language model has experienced a transformation from statistical model to neural network model and then to the deep neural network model. The wide application of depth learning makes language modeling more extensive, complex, and expensive. This paper combines the personalized input, convolutional neural network (CNN) coding, and the technique of union gate, cooperating with long short-term memory (LSTM) mechanism to improve the language model. The dynamic integration of LSTM and CNN is called Gated CLSTM. In the experiment, we used the deep learning framework Tensorflow to achieve a Gated GLSTM architecture. Besides, some classical optimization techniques, such as noise contrastive estimation and re-

收稿日期：2017-06-25；修回日期：2017-12-10

基金项目：国家社会科学基金“施引者引用意向与文献计量视角的学术论文被引影响因素研究”（17BTQ014）。

作者简介：王毅，男，1992年生，硕士研究生，主要研究方向为自然语言处理；谢娟，女，1995年生，博士研究生，主要研究方向为信息计量；成颖，男，1971年生，教授，博士生导师，主要研究方向为用户信息行为、信息检索，E-mail: chengy@nju.edu.cn。

current projection layer, were adopted in the experiment. We tested the performance of the Gated CLSTM under an open and big scale corpus set and trained a signal-layer model and a three-layer model to observe how network depth influences the performance. The single-layer model has 4 days of training experience and reduced the perplexity to 42.1 in four GPU console environment. The three-layer model reduced the perplexity to 33.1 in 6 days. Compared with some classical benchmark models, significant improvements have been made by Gated CLSTM considering both hardware and time complexity and perplexity.

Key words: language model; recurrent neural network; convolutional neural network; character sequences encoding

1 引　言

自然语言处理的研究史，贯穿着学界建构能完美表示、计算、理解和生成语言模型的尝试。早期的模型都是在受限的领域由人工基于规则构建，由于自然语言的复杂性和多样性，基于规则的方法不足以独立解决自然语言处理中的核心问题。随后，基于统计的研究者在大规模未标注的语料中通过统计学习语言的语法、语义和语用信息，试图构建一个能够深度理解自然语言的模型。至今，基于统计的语言模型已经历了从统计模型、神经网络模型到深度神经网络模型的衍化。

第一阶段的统计语言模型有生成模型与判别模型之分，前者的目标是对一个词项（term）序列的联合概率分布进行建模，该序列可能是一个单词、一个句子或者整个篇章。生成模型借助于贝叶斯法则可以转换为判别模型，为词项序列下一个可能出现的词项指派概率。概率越高表示该序列越符合语言规则，它的出现也就越“合理”。过去三十年，学界相继提出了多个不同的语言模型，其中 n-gram 模型最具代表性。对 n-gram 进行改进的高级语言模型也相继出现，比如，基于决策树和最大熵的模型中加入了诸如词性及语法结构等特征。以 n-gram 为代表的统计语言模型具有时间复杂度低、实现简单且可靠等优点，但也存在着长距离依赖、模型泛化能力弱[1]、维度灾难以及语言表示能力差等缺陷。

直到 Bengio 等[2]实现了前馈神经网络语言模型，开启了语言模型第二阶段的研究，该阶段算法上的主要变化是由原先的上下文计数（context-counting）转变为上下文预测（context-predicting）[3]。实现层面的主要变化是统计的基本单位由词项转变为词向量（word vector），从而使计算机能够更有效对其进行识别和计算。相比统计模型，神经网络模型能够接受相对更大的上下文，缺点是模型架构与语言数据的序列特性不吻合，模型训练的复杂度高，在研究与应用中通常需要限定词表和语料的规模。

深度学习在自然语言处理中的应用引发了语言模型第三阶段的研究，作为深度神经网络家族的一员——循环神经网络（recurrent neural network，RNN）具有图灵完备性和序列建模的特点，使其成为语言建模不可或缺的方法；RNN 的缺点是存在梯度消失和梯度爆炸两个严重的不足，同时大规模循环神经网络的训练速度较前馈神经网络模型没有优势，对硬件要求更高。针对深度神经网络的不足，学界多从模型架构、训练时间、隐藏层的配置以及输入长度等角度进行了优化，其中后两者属于工程领域的诀窍（trick），学术论文中通常着墨不多。综合不同的优化策略可以发现：语言模型正处于多技术融合的阶段，从单一维度的优化不足以显著提高模型的整体性能。

针对先前研究中字符信息利用不充分的不足，本文模拟人类阅读文献时并行处理字符序列与词项的做法，将词向量与字符一并输入模型，提出了模型输入拟人化的思路；相较于 RNN，CNN 更适合于提取局部特征，据此本文采用 CNN 对没有明确语法规则的字符序列进行编码；对于编码后的字符序列与词向量的融合问题，本文借鉴了 LSTM 中输入门以及遗忘门的特点，提出了融合门机制。本文利用深度学习框架 Tensorflow 实现了 Gated CLSTM。实验环节还采用了负采样及循环投影层等经典的优化技术，在包含近十亿个英文单词的通用数据集（one billion word benchmark）下测试了模型的性能，取得了满意的效果。

2 相关研究

2.1 词向量

深度学习是语言模型第二以及第三阶段的核心技术。Bengio 等[4]将神经网络技术应用于高维离散数据的联合概率分布建模，用于解决维度灾难问题。基于此，Bengio 等[2]实现了前馈神经网络语言模型（NNLM），实验结果表明该模型优于传统的 n-gram

模型。Mikolov 等[5]的研究删除了 NNLM 中的隐藏层，提出了 Continuous Bag-of-Words（CBOW）和 Continuous Skip-gram Model（Skip-gram）两个简化的前馈神经网络语言模型，二者以及相应的优化算法统称为 word2vec 算法，是目前训练词向量最高效的工具之一。词向量表示了语言的深层语义，用稠密的向量解决了传统 one-hot 表示带来的维度灾难和词汇鸿沟问题[6]，能够有效地对单词之间的语义进行计算。Le 等[7]在词向量的基础上试图构建句子和短语的向量表示，克服词袋模型的缺点，以提高信息检索的性能。词向量存在的缺陷主要有：基于词项为"单义词"的假设、难以进行词义消歧[8]、低频词的学习效果较差[9]以及缺乏更合理的评价指标等。尽管词向量在词项的表示方面已经取得了明显的进步，但仍然难以完美的表示词项的所有信息，因此仅输入词向量的模型没有充分利用接收到的所有有用信息，如输入中的字符序列信息等。

2.2 循环神经网络

目前，前馈神经网络模型多用于词向量的训练[10]，其在语言模型中的应用则存在着仅接收固定长度的输入、缺乏"记忆"和"遗忘"机制、假设词项独立以及不符合序列数据建模等缺陷。Cohen 等[11]最早提出了用于序列数据监督学习的架构，将前馈神经网络中的输出神经元连接到具有自连接特征的特殊神经元（special units），并将其作为输入加入下一个步骤的隐藏层计算。Elman[12]在 Cohen 等工作的基础上提出了一个简单的循环神经网络架构，将隐藏层中的神经元定义为语境神经元（context units），将各步骤之间的语境神经元进行了连接，即语境神经元存储了上一步骤计算后"记忆"信息。RNN 能够灵活的利用语境信息、有效地学习到"记忆"机制，可以接收不同类型数据以及在序列数据失真的情况下识别出序列模式[13]。Mikolov 等[14]进行了探索性的实验研究，将 KN5（modified Kneser-Ney smoothed 5-gram）和不同配置的 RNN 语言模型应用于语音识别，采用混淆度（perplexity，PPL）和词错误率（WER）作为评价指标，获得了一组循环神经网络语言模型（RNNLM）的测试结果，数据表明 RNN 的综合性能超过 KN5。随后，Mikolov 等[15]又在 Penn Treebank 数据集上完成了拓展实验。结果显示，自适应的 RNNLM 模型可以将混淆度降至 101.0。上述研究表明，RNN 具备能够更好地利用序列信息、能够"记忆"神经元之间传递的信息、能更深层次地揭示数据中隐藏的模式等特点，使其在语言模型研究中有着举足轻重的地位。不过，RNN 也存在诸多缺陷，比如，Bengio 等[16]发现模型训练过程中存在长距离依赖的学习困难问题，Hochreiter 等[17]的研究显示 RNN 存在梯度消失和存储"记忆"信息不稳定等问题。Lipton 等[18]肯定了 RNN 在序列建模中的价值，分析显示虽然 RNN 的训练需要大量的计算资源且时间复杂度较高，但因为 LSTM[19]、最优化算法以及并行计算等优化策略的出现，使其能够在多项序列建模中取得更优的效果。语言模型显然是一个序列建模问题[20]，Lipton 等[18]的工作提示结合多项优化技术的 RNN 架构是语言建模的一个有效路径。

2.3 输入与编码

根据输出端预测的结果可以将语言模型划分为词项级别和字符级别。现有研究中的语言模型多为词项级别[21-23]，也有少量研究着眼于字符级别。比如，Karpathy 等[24]利用字符级别的模型展示了 LSTM 在长距离依赖上的学习能力，Ballesteros 等[25]用字符替代词项构建了 LSTM，提高了依存分析的精确度。字符级别的模型多应用在词性标注及依存分析等特定的 NLP 任务中，用于补充词项信息的不足。根据语法和人类习得，词项是含有语义或语用信息的最小语言单位，以词项为中心的语言建模更符合语言的生成过程，Karpathy 等[24]的研究表明字符信息可以作为词项的补充，提高模型的整体性能。据此，在输入端便存在三种信息的输入方式：词向量、字符向量以及二者同时输入。当前的研究仍然以输出端的目标来限定输入端的信息选择，构建诸如"词项-词项"和"字符-字符"的模型，将"词项字符混合-词项"作为建模的目标将是有益的尝试。

RNN 和 CNN 是目前最常用的两种语言编码器，以 RNN 为基准架构的模型适合于针对词项的序列结构信息建模，例如，Cho 等[26]用 RNN 作为翻译模型的编码器和解码器，学习源语言到目标语言的序列转换过程。Sutskever 等[27]则是用多层的 LSTM 作为编码器和解码器，学习短语和句子的表示。RNN 作为序列数据的编码器取得了令人满意的效果，编码有效的提取出了序列生成的规则信息。与 RNN 不同，CNN 不关心序列中元素的整体顺序关系，而更注重提取出深层次的局部特征，常在分类任务中作为语言的编码器出现。Kalchbrenner 等[28]利用 CNN 对句子建模，表明不依赖于解析树也可以有效的对句子编码，且 CNN 对任何语言都适用。Kim[29]将 CNN

用于句子分类，详细阐述了对语言卷积的细节，从算法角度分析了 CNN 的编码特点。Tang 等[30]针对文本建模采用了两层神经网络来学习文本表示，即先利用 CNN 学习句子表示，再用双向神经网络学习文本表示，通过情感分类的标签作为监督信息，以关联文本的语义。CNN 在 NLP 的应用可以看出，CNN 更类似于人类的视觉，把目光聚焦于局部信息，最后通过局部特征组合做综合判断，目光的移动也没有特定的顺序要求。

综上所述，RNN 是一个顺序结构，适合于序列特征编码；CNN 是一个层次结构，适合于局部特征编码。探索 RNN 与 CNN 的结合，即采用 RNN 处理词项序列，CNN 处理词根、前缀和后缀等局部特征的字符序列，将使模型能够更有效地利用这两类信息，收模型优化之效。

3　Gated CLSTM

3.1　架构设计

神经网络语言模型大多是单词到单词（word-to-word）或者字符到字符（character-to-character）的模型。少量研究针对字符到单词（character-to-word）建模，Wang 等[31]提出了一个字符到单词的模型，称为 C2W，接受字符向量作为输入，利用双向循环神经网络输出单词的分布式表示。近期，也出现了字符和单词混合输入的研究。比如，Kang 等[32]在前馈神经网络语言模型中就使用了该方法，单词向量和字符向量通过连接操作融合，作为历史信息用于预测，没有对两方面信息做区别处理。Dos Santos 等[33]认为词向量不足以反映出词形的重要性，而词形是词性标注的重要特征，因此他通过字符向量补充模型的词形信息，同时联合词向量完善了模型，在词性标注上取得了 97.32%的精确度。Bojanowski 等[34]拓展了字符到字符的模型，加入了词项的信息，以词向量为条件预测字符。Luong 等[35]在机器翻译模型中采用了字符和词的混合输入，用于解决未登录词的问题，用字符预测替换未登录词的特殊符号 UNK。

研究问题 1　模型输入拟人化。上述研究可以发现字符信息在 NLP 任务中都能起到应有的作用。不过，少量的研究中即使同时输入了二者，也未能对其进行有效处理。比如，连接操作没有合理的依据[32]、不符合人类思考的习惯[22]、主要集中于应用[21,23]、未解决语言模型本身的问题等。模拟人的思考与行为一直是人工智能研究的主要方法。具体到文献阅读，可以发现人类在视觉上既看到了单词，也看到了组成单词的字符，不过人类对二者进行了不同的处理。为了更好地模拟人类的阅读行为，本研究认为语言模型的预测应该同时关注词项内部的结构（如前缀、后缀以及词根等）和词项自身的整体特征。据此，本研究将字符与单词一并作为模型的输入，完成语言模型的建构。

研究问题 2　卷积神经网络编码。显然，词项和字符表达了不同层次的信息，Kang 等[20]采用的向量连接操作把二者作为同一层次的信息处理不符合基本的语言规则，也无法反映出二者的区别。本研究通过对字符序列编码之后提升了信息表达的层次，使其具有了词项信息的内涵。语言学上，可以通过词法规则规约词之间顺序[36]，而字符之间的组合不存在类似规则。从字符序列到序列特征向量有一个编码过程。根据词项结构（即词根、前缀及后缀等）的层次特征，本文引入了擅长于抽取局部特征的卷积神经网络（convolutional neural network，CNN）对字符序列进行编码，进而获得字符编码后的特征向量。

研究问题 3　融合门机制。沿着上述两项工作，对特征向量和词向量进行信息融合是自然的后续处理。本研究在 LSTM 中增加了融合门机制，从而能够在经典的遗忘门、输入门以及输出门等门机制的基础上更加有效的利用字符序列中隐藏的语法和语义信息。

本文通过上述 3 项研究工作结合长短时记忆单元（long short-term memory, LSTM）优化了语言模型，提出了结合 LSTM 和 CNN 混合架构的深度神经网络语言模型（Gated CLSTM），Gated CLSTM 可以进行堆积，形成深层的神经网络，模型架构如图 1 所示。

图 1a 和图 1b 是经典的 RNN 语言模型架构。图 1a 表示输入字符 T″，用 LSTM 循环传递信息，预测字符 H；图 1b 表示输入单词 THE 的词向量，预测下一个词 CAT；图 1c 为本文改进的模型，在输入时分别输入字符 T、H、E 的字符向量，利用 Char CNN 对字符序列编码，然后传入 Gated LSTM。用 CNN 编码就能充分的提取到内部的结构特征，从而有效地利用更多信息。

3.2　建模过程

Gated CLSTM 的内部架构如图 2 所示，具体分为 4 层：输入层、卷积和嵌入层、Gated CLSTM 层和输出层。

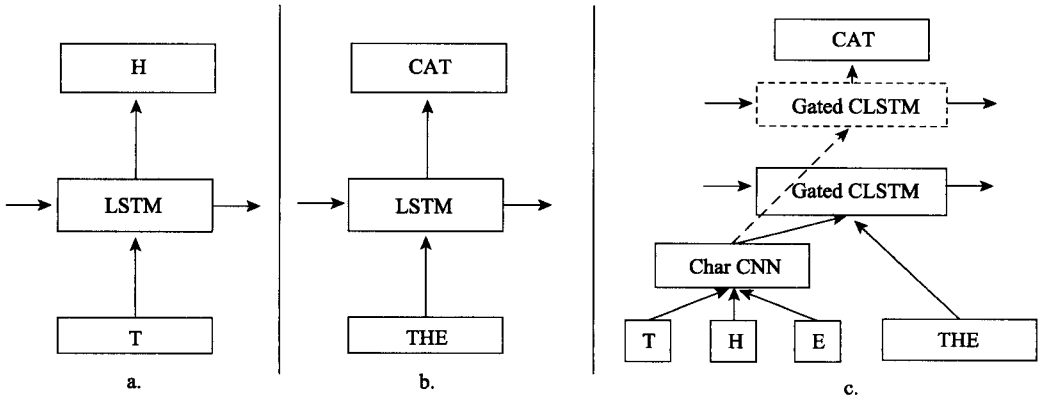

图 1 模型架构

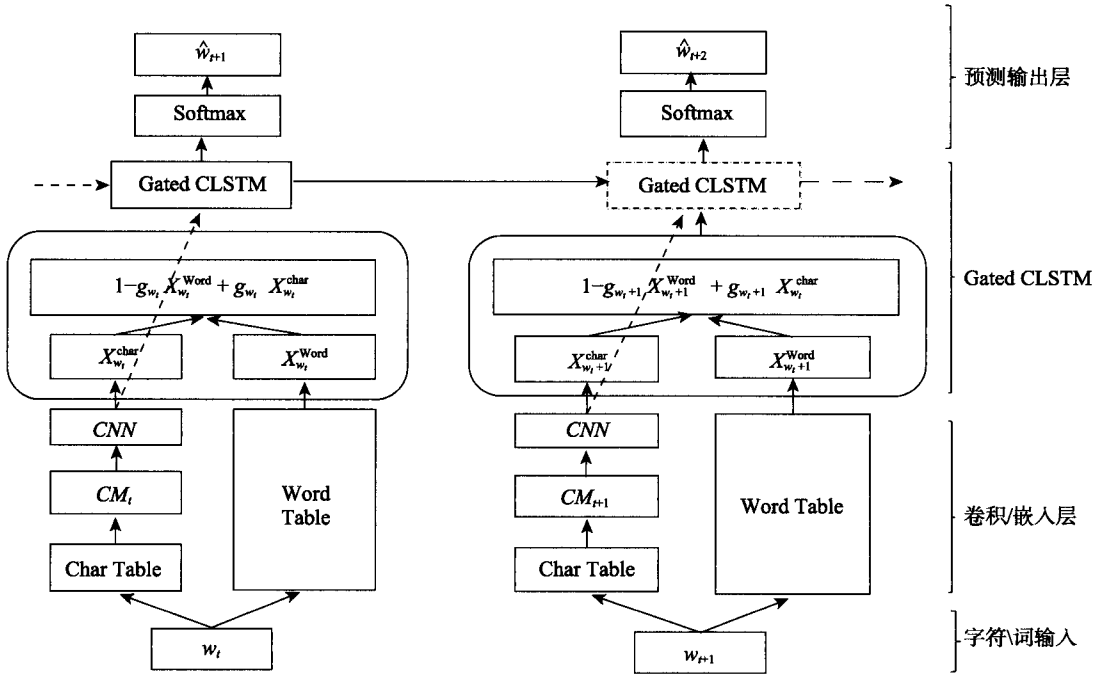

图 2 Gated CLSTM 内部架构

3.2.1 输入层

输入层接收的信息是一个词项序列 $\{w_1, w_2, \cdots, w_n\}$，第 t 个步骤的输入单词用 w_t 表示。每个词项都有一个对应的字符序列，例如，w_t 对应着字符序列 $\{c_{t1}, c_{t2}, \cdots, c_{tm}\}$，其中 n 和 m 分别表示模型接受的词项序列和字符序列的长度，它们属于超参数的一部分，由数据集和工程实现决定。

句子和词项长度都是可变的，变长的输入是 RNN 模型在工程上的一个难点。实现环节可以将长度作为一个可变参数，建立循环，完成样本遍历。此外，循环的实现又涉及诸如自动梯度计算等一系列问题，因此，变长的 RNN 实现困难且难以保证效率。虽然，RNN 在数学上可以处理任意长度的数据，但考虑到实现的效率问题，等长数据的矩阵批处理运算有速度的绝对优势，因此实现中常将变长序列通过填充法（padding）处理为等长数据，其他方法还有装桶法（bucket）[37]及动态 RNN[38]。填充法是一种数据预处理方法，即首先设定一个最大长度，在不满足长度的序列后附加上特殊的填充符号（PAD）。输入长度的优化依赖于数据分布情况、经验选择以及高效的底层实现。由于语言模型容易过拟合，使其倾向于预测出填充符号，降低了模型的泛化能力。

合适的填充长度可避免浪费计算资源，本文在输入层结合实验数据集和前人经验对 n 和 m 两个超参数分别设为 20 和 16。利用填充法使数据变为等长的序列，再对序列开始和结尾添加特殊的符号标识。

输入层在 t 步骤将词项 w_t 和字符序列 $\{\text{char}_{t1},\text{char}_{t2},\cdots,\text{char}_{tm}\}$ 传入卷积和嵌入层。

3.2.2 卷积和嵌入层

卷积和嵌入层分别处理字符序列和词项。卷积层的作用是对 $\{\text{char}_{t1},\text{char}_{t2},\cdots,\text{char}_{tm}\}$ 编码，提取出字符序列的特征向量。嵌入层的作用是根据 w_t 在 Word Table 中通过索引找到对应的词向量 $X_{w_t}^{\text{word}}$ [公式(1)]，Word Table 表示嵌入矩阵。

$$X_{w_t}^{\text{word}} = \text{Lookup}(\text{WordTable}, w_t) \tag{1}$$

卷积神经网络编码的结构由一个卷积层和一个最大池化层组成，每个序列的最大长度是 m。字符向量的维度大小用 d 表示。将一个词表示成为字符矩阵 $CM_t \in \mathbb{R}^{d\times m}$。卷积操作涉及一个过滤器 $W_c \in \mathbb{R}^{d\times h}$，$h$ 表示过滤器移动的窗口大小。一个过滤器卷积生成特征向量可通过公式(2)计算。

$$c = f\left(\text{conv}\left(X * W_c\right) + b\right) \tag{2}$$

其中，f 表示非线性的激活函数，conv 表示卷积过程。生成了一个特征向量 $cf \in \mathbb{R}^{m-h+1}$，$b$ 表示的是偏置向量。通常对文本的卷积可以设置多个不同大小的过滤窗口，如使 $h=[2,3,4]$。每个长度也可以设置多个过滤器，假如对每个长度设置两个过滤器，那么整个卷积将产生 $x=6$ 组特征向量，用 $[cf_1,cf_2,\cdots,cf_x]$ 表示。然后对每一组特征向量做最大池化操作[公式(3)]，生成字符序列的编码特征向量：

$$x_{w_t}^{\text{char}} = \left[\max\{cf_1\},\max\{cf_2\},\cdots,\max\{cf_x\}\right] \tag{3}$$

然后，将 $x_{w_t}^{\text{char}}$ 和 $X_{w_t}^{\text{word}}$ 传入 Gated CLSTM 层。

3.2.3 Gated CLSTM 层

Gated CLSTM 层在 LSTM 的基础之上加入了融合门，用于对字符编码特征向量和词向量进行融合，用门机制来决定信息保留的程度。由于 $x_{w_t}^{\text{char}}$ 和 $X_{w_t}^{\text{word}}$ 的维度不一致，先对 $x_{w_t}^{\text{char}}$ 做一个投影操作[公式(4)]，使其维度一致：

$$x_{w_t}^{\text{encoding}} = W_p x_{w_t}^{\text{char}} + b_p \tag{4}$$

其中，$x_{w_t}^{\text{encoding}}$ 表示投影后的字符序列编码特征向量，然后计算融合门[公式(5)、公式(6)]：

$$g_{w_t} = \sigma\left(W_{ge}x_{w_t}^{\text{encoding}} + W_{gw}x_{w_t}^{\text{word}} + b_g\right) \tag{5}$$

$$x_{w_t}^{\text{combine}} = \left(1-g_{w_t}\right)x_{w_t}^{\text{word}} + g_{w_t}x_{w_t}^{\text{encoding}} \tag{6}$$

其中，W_{ge}，W_{gw} 和 b_g 表示参数和偏置项，g_{w_t} 是经过 sigmoid 运算生成的值在 0 和 1 之间的向量，用于决定哪些信息应该融合，$x_{w_t}^{\text{combine}}$ 表示融合后的向量，然后计算遗忘门，遗忘门用于决定哪些信息需要移除。

$$f_t = \sigma\left(W_{fx}x_{w_t}^{\text{combine}} + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f\right) \tag{7}$$

c 和 h 存储了一个序列的上下文信息，上一个步骤的单元状态和隐藏状态分别表示为 c_{t-1} 和 h_{t-1}。然后计算输入门：

$$i_t = \sigma\left(W_{ix}x_{w_t}^{\text{combine}} + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i\right) \tag{8}$$

输入门用于决定哪些新的信息需要加入到当前单元状态中，用 sigmoid 函数输出一个选择向量 i_t 实现[公式(8)]。当前单元状态的计算见公式(9)：

$$c_t = f_t \odot c_{t-1} + i_t \odot \tan h\left(W_{cx}x_{w_t}^{\text{combine}} + W_{ch}h_{t-1} + b_c\right) \tag{9}$$

其中，c_t 表示当前步骤的单元状态，存储了遗忘和更新后的上下文信息。然后，进入输出门，计算 Gated CLSTM 需要输出的信息：

$$o_t = \sigma\left(W_{ox}x_{w_t}^{\text{combine}} + W_{oh}h_{t-1} + W_{oc}c_t + b_o\right) \tag{10}$$

$$m_t = o_t * \tan h\left(c_t\right) \tag{11}$$

输出门决定了需要输出哪些信息，用于当前步骤的预测[公式(11)]。输出选择向量 o_t 也采用 sigmoid 函数进行计算[公式(10)]。在普通的 LSTM 中，直接输出 m_t 到下一层即可。本文采用了循环投影层（recurrent projection layer）[39]的结构，用于有效地增加模型记忆单元的大小，同时减少循环链接和输出门的参数数量[公式(12)]。

$$h_t = W_{hm}m_t \tag{12}$$

其中，W_{hm} 是参数矩阵，最后 Gated CLSTM 将 h_t 传入输出层预测当前步骤的下一个词项，将 c_t 和 h_t 传入下一个步骤的 Gated CLSTM 作为上下文信息。

3.2.4 输出层

输出层用于计算当前步骤预测的概率：

$$h_{to} = W_o h_t + b_o \tag{13}$$

$$y_t = \text{full\_softmax}\left(h_{to}\right) \tag{14}$$

其中，y_t 是一个和词表大小维度一致的向量，每一维表示了对应词项成为预测词的概率。Full_softmax 表示一个完全的 softmax 公式，对概率进行了归一化。训练阶段，softmax 的归一化是计算代价极其高昂的操作，大词表的数据集尤甚。

针对 softmax 时间复杂度高的问题，Bengio 等[2]早在 2003 年就提出用重要性抽样的方法来近似估计梯度。重要性抽样是一种近似估计方法，通过选择一个合适的概率密度函数，称为提议函数（proposal

function)来替代不易获得的真实函数进行抽样估计，因此，提议(proposal)的选择至关重要。Bengio等[40]用Brown语料库中生成的unigram分布作为提议，他认为固定的unigram不适应于持续的模型训练过程，由于unigram分布和实际分布的偏差会随着训练迭代次数的增加而增大，随之会造成梯度估计的方差增大，表现为模型的不稳定。Gutmann等[41]提出的NCE(Noise Contrastive estimation)方法为没有归一化的统计模型提供了一种参数估计方法。Mnih等[42]把NCE应用到语言模型，证明了NCE能够在不影响模型效果的前提下减少一个数量级的训练时间。Dyer[43]阐明了NCE是一种对任意局部归一化语言模型都有效的参数估计方法。根据Gutmann等[41]的研究结论，本文采用NCE对模型的训练时间进行了优化。

4 实验部分

4.1 实验过程

4.1.1 数据描述

本实验采用的数据是包含近十亿个英文单词的数据集(one billion word benchmark)，由Chelba等[44]在论文中发布，是一个可公开易获取的基准数据集，专门用于评价语言模型的性能。该数据集提供的词表总共有793471个词项，其中包括句子边界等符号。所有的句子都随机打乱且移除了重复的句子。未登录词大概占比0.3%，用特殊的符号“UNK”替代。数据集主要的来源是WMT11网站，经过了比较常规的文本预处理清洗，包括训练集和测试集两部分。选择该数据集主要基于以下考虑：公开易获取，数据已经经过有效的预处理，已在多个模型上进行了测试。

4.1.2 模型配置

实验的评价指标为混淆度，在训练集和测试集上同时进行。为了避免句子长度差异带来的问题，用特殊的标识符附加在句子首尾，$和^分别表示句子的开始和结束，避免了填充方法带来的计算资源浪费。相当于让模型自动地去学习到一个句子的结束和开始。对于单词的输入使用填充法，即在每个单词首尾添加标识符，例如，cat处理后变为$cat^，然后用零填充到最长的词项长度$m$，用于CNN模型的输入。实验的超参数选择如表1所示。

表1 超参数的选择

参数	数量	解释
batch_size	256	批量大小，表示单词训练输入的样本数量
num_steps	20	步长，表示RNN展开的长度，即上下文长度
word_length	16	最长单词长度，表示词表中最长的单词长度
num_layers	1~3	神经网络层数，表示模型的深度
learning_rate	0.5	学习速率
max_grad_norm	10.0	最大梯度标准值
keep_prob	0.8	dropout的比例
vocab_size	793470	词表大小
char_size	128	字符表的大小
emb_char_size	32	字符向量的维度
emb_word_size	512	词向量的维度
state_size	2048	隐藏单元的维度
projected_size	512	投影向量的维度
num_sampled	8192	负采样的个数

其中，部分超参数主要来自于先前论文研究中的经验，如学习速率、用于控制梯度爆炸的最大梯度标准值和dropout比例。一些参数是由数据集的特性而设置，如词表的大小、字符表的大小、最长单词长度等。另外还有部分参数是根据LSTM的深度和硬件的条件配置，如批量大小、GPU个数。本文分别实验了一层和三层两个不同深度LSTM模型的效果，验证Gated CLSTM的有效性。在大规模的数据集上训练避免不了考虑分布式的使用，本文使用在4个Titan X GPU之上并行运行，采用了同步式的梯度下降方法。模型的超参数全部采用表1的配置。针对大规模词表(793470个单词)，模型采用了NCE方法，加快训练速度，每次梯度估计从词表中抽样8192个样本完成。

4.2 实验结果分析

本文对实验的结果进行翔实的展现和分析，对改进模型和处于前沿的基准模型的性能进行了比较。由于外部指标通常涉及外部的数据任务及对模型架构的更改，不可控因素较多，因此本实验结果仅依据模型的内部评价指标混淆度展开分析：

$$\mathrm{PPL}=P\left(w_1 w_2 \ldots w_N\right)^{-\frac{1}{N}}=\sqrt[k]{\prod \frac{1}{P\left(w_i \mid w_{1 \ldots i-1}\right)}}=\mathrm{e}^{-\frac{1}{K}\sum_{i=1}^{K}\log_{\mathrm{e}} P\left(w_i \mid w_{1 \ldots i-1}\right)} \tag{15}$$

4.2.1　训练过程

实验分别在单层模型和三层模型上进行了大约 1 周的训练，图 3 展示了三层模型从 0 步到 80 万步在测试数据上的表现。

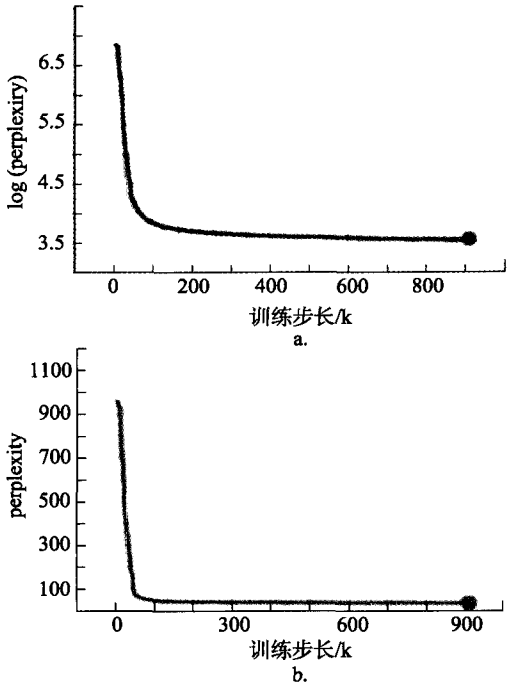

图 3　三层模型的训练过程

图 3 中横坐标表示训练的步数，图 3a 反映了对数混淆度（log perplexity）的变化情况，图 3b 反映的是混淆度（perplexity）的变化情况，根据公式（15），对数混淆度就是一批样本的平均损失。从基本的趋势来看，模型在训练开始后学习速度很快，损失和混淆度都迅速下降，在大约 20 万步后，曲线趋向于平稳，其后的 60 万步对数混淆度在 3.8~3.5 稳中有降，对应的混淆度则是在范围 44.7~33.11。从数值的效果来看，混淆度可以更直观地反映差距，例如,平均损失 3.5 对应的混淆度 33.11，3.6 对应混淆度 36.60，因此模型的真实差距并没有混淆度数值的差距那么大。模型大部分时间是在一个很小的范围内逐渐收敛，学习的速率越来越慢。为了更清楚地观察模型指标的变化，图 4 所示是单层模型从 20 万步至 100 万步的训练过程。

由图 4 可以看出，单层模型的对数混淆度在 3.74~3.78 波动，说明模型在此区间已经基本收敛，对应的最低混淆度是 42.09。对比前面的结果，三层模型的对数混淆度在 3.5 附近收敛，对应的混淆度为 33.11，因此三层模型能够具有更好的学习效果。

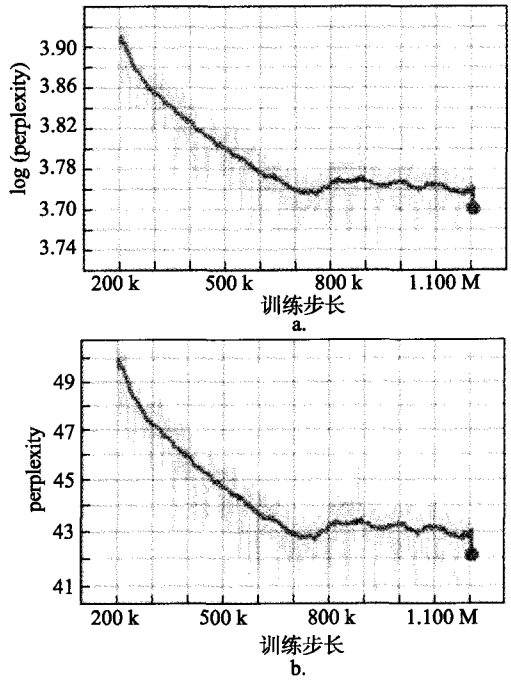

图 4　单层模型 200 k~1100 M 步模型的训练过程

4.2.2　模型的性能分析

本文选择了 7 种近些年综合表现最为优秀的语言模型进行了对比。与本文相同的是，其数据集都是“one billion word benchmark”，保证了可比性（表 2）。

表 2　模型的性能比较

模型	混淆度	硬件	时间
SIGMOID-RNN-2048	68.3	1 CPU	10 天
Interpolated KN 5-gram	67.6	100 CPUs	3 小时
RNN+KN-5	42.4	1 GPU	14 天
LSTM-2048-512	43.7	32 GPUs	5 天
LightRNN	66.0	1 GPU	10 小时
GCNN-13	38.1	1 GPU	14 天
BIG LSTM+CNN INPUTS	30.0	32 GPUS	21 天
1-Layer Gated CLSTM	42.1	4 GPUs	4 天
3-Layer Gated CLSTM	33.1	4 GPUs	6 天

下面本文将从改进思路、性能、硬件环境及训练时间等角度展开具体的分析。

1）统计模型与 RNN

目前，第一阶段统计模型的优秀代表 Interpolated KN 5-gram 已经成为评价第二以及第三阶段语言模型的基准，插值法和 Kneser-Ney 平滑是其核心技术，通过 100 个 CPU 的并行训练，历经 3 小时混淆度降到了 67.6[44]。第二阶段的代表 SIGMOID-RNN-2048[45] 在经典 RNN 的基础上提出了一种称为 BlackOut 的近似算法以适应大规模词表的训练，其实质是在输

出层应用了 DropOut 机制优化了大规模词表全概率估计的难题。该模型的优点是不需要 GPU 的支持，经过 10 天的训练使混淆度降到了 68.3。

从训练时间上看 5-gram 有优势，但其需要大量的 CPU 资源（100 颗）。Interpolated KN 5-gram 模型的不足表现为上下文距离只有 5，无法解决长距离依赖；当距离增加时，算法的时间复杂度会急剧上升。而改进的 RNN—SIGMOID-RNN-2048 模型能仅用一颗 CPU 即可达到 5-gram 的混淆度，能够携带更长的上下文信息，且能够对文本进行嵌入式表示，因此基于深度学习的语言模型整体上优于统计模型。

RNN+KN-5[46]是前两个模型的集成，其在单个 GPU 上经过 14 天的训练可把混淆度降低到 42.4，相比单一的模型降低了大约 25 个点，取得了显著的提高。由实验结果可见，统计模型与 RNN 单独建模很难取得良好的混淆度，而两项技术的综合运用则可以显著提高模型性能。

2）深度神经网络模型——高成本

相较于 SIGMOID-RNN-2048，LSTM-2048-512[47]在 LSTM 的基础上加入了循环投影层，2048 表示中间隐藏单元的维度，512 表示的是投影的大小，利用了 32 个 GPU 经过 5 天的训练使混淆度降低至 43.7，证明了 LSTM 和循环投影层的结合能够有效降低混淆度。该模型的训练效果还说明了在大规模数据下，如果有足够的硬件支持和训练时间，LSTM 模型能够达到比较好的效果。

BIG LSTM+CNN INPUTS[47]是一个结合字符 CNN 编码作为输入的 LSTM 模型，该模型在 32 个 GPU 的硬件环境下训练了 21 天，接近了已经报告的语言模型混淆度的最优值 30.0，其中 BIG 是指隐藏层的维度为 8192、投影层大小为 1024 的循环投影网络。相较于 LSTM-2048-512，混淆度降低了 31.4%，是一个非常显著的进步，在新近的成果中，有研究可以把混淆度再次降低，但大规模神经网络训练存在的过拟合问题及频次分布不均匀加重的过拟合问题，使得细微的降低已很难分清孰优孰劣。

3）深度神经网络模型——低成本

目前，普通的 PC 机主板最多支持 4 块显卡，LSTM-2048-512 与 BIG LSTM+CNN INPUTS 模型的实验环境所要求的 32 块显卡，需要多个服务器组成的集群来运行，所以硬件条件目前仍然限制着神经网络语言模型的应用与普及。另外，深度学习模型并不是一种适合并行计算的模型，因此大规模的分布式机器学习仍是目前研究的热点和难点。

在这种背景下，一些研究通过优化使得模型能够在廉价的硬件和可接受的训练时间下运行。LightRNN[48]来自于微软研究院，通过压缩词表使模型的时间复杂度显著下降，在单 GPU 机器上仅用了 10 个小时就将混淆度降低到 66.0，相较于 SIGMOID-RNN-2048 模型的 68.3 没有取得显著提升，但是时间仅是其 1/24。该模型甚至能够在移动设备上运行，极大地节约了硬件和训练成本。

GCNN-13[49]模型源于 Facebook AI 实验室，基于 CNN 提出了一种新颖的门机制使模型能够适应于序列输入，13 表示了一个 13 层的卷积网络，在单个 GPU 机器上经过 14 天的训练使混淆度降低到了 38.1，在混淆度与硬件成本上都超过了 LSTM-2048-512 模型，这是非循环方法在语言模型任务上首次超过循环模型。

本文的两个模型硬件环境与 LightRNN 和 GCNN-13 相近，都可以在一台 PC 上进行，1-LAYER Gated CLSTM 在混淆度与 GCNN-13 伯仲的情况下，显著降低了时间复杂度，所需时间仅是后者的 1/3 左右，3-LAYER-Gated CLSTM 相较于 GCNN-13，混淆度下降了 13%，所需时间仍是后者的 1/2 左右；相较于混淆度的标杆 BIG LSTM+CNN INPUTS，3-LAYER-Gated CLSTM 以低廉的硬件配置仅落后 10%左右，但是所需时间仅是其 1/3。

综上所述，语言模型的研究是架构、硬件和时间等的折中，本文改进得到的模型是一个相对综合、集成的模型，能够在相对廉价的硬件基础之上，经过可接受的训练时间，学习出一个高性能的模型。

5 结论与讨论

本文对深度学习技术在语言模型上的相关研究进行了深入分析。围绕当前语言模型存在的不足和应用的困难对现有模型进行了改进与优化，构建了一个轻型的深度网络语言模型，该模型能够在 PC 平台上对大规模语料开展语言模型的理论与应用研究。通过混淆度、硬件以及时间三个维度的综合比较，本研究构建的 3-LAYER-Gated CLSTM 模型具有优异的性价比。根据实验数据，本研究对深度神经网络语言模型研究的结论与启示有：

（1）**综合集成是语言模型完善的必由之路。** RNN+KN-5 整合了两个技术流派显著降低了混淆度；BIG LSTM+CNN INPUTS 集成了 LSTM 和 CNN 之后，实现了性能的飞跃；本研究中两个模型多项优化技术的综合运用实现了优异的性价比。这些研

究都说明在语言建模中不同技术流派、不同优化技术的有机结合能显著地提高模型的综合性能。

（2）**字符与词项信息综合应用的有效性**。建模的首要步骤是对数据的观察和假设，通过已有的实践和对语言数据的认知，语言模型的输入信息不应该在词项和字符之间二选一，而是缺一不可。原因有二，一是这符合人类接受语言信息的真实情境，即在接收到词汇的同时也接收到了字符信息；二是在预测过程中，字符序列中的前缀、词根或后缀都对预测起作用，因此字符是有效的语言建模信息。

（3）**字符和词项处理需采用不同的处理方式**。字符和词汇不是一个层次的信息，不能够通过向量连接这样简单的操作对其融合，需要先对字符序列编码。由词法规则可知，词汇能够用语法规则来解释其间的相关性，而字符序列没有明确的规定，两个信息需要不同的机制进行处理。据此，本文引入了卷积神经网络对字符序列进行编码，然后通过门机制融合两方面信息。实验结果表明，该思路显著提高了模型的性能。

（4）**深度语言模型仍然是未来研究的方向**。统计语言模型具有明显的缺点，表现为不能有效地表示语言、长距离依赖及模型泛化能力差，神经网络模型以及深度神经网络有效地改善了上述问题。从建模思想与性能提升的角度来看，基于深度学习技术的语言模型将是未来发展的方向。

（5）**RNN 仍然是语言模型建模的基础**。众多的实验结果证明了其在序列建模上的优势，RNN 建模思想符合语言的生成过程，对于其存在的诸多问题可以通过合理地融合多种优化技术加以解决，比如本研究采用的字符序列与词项的拟人化输入、CNN 编码、循环投影层及负采样技术等。

（6）**宜综合权衡科学性与可行性**。语言模型是数学和工程的折中，由于深度学习技术在工程领域仍然存在着许多黑箱与技术诀窍（trick），因此，语言模型需要在工程实践中积累大量的经验。语言模型硬件环境存在较高程度的依赖，其训练常需要数天甚至数周，因此，应充分考虑可行性，而不能一味地提高模型的复杂度。

本文的研究也表明，除了已经进行的工作之外，语言模型还可以在以下几个方面进行更深入的优化与改进：

（1）**语言模型有必要加入更多的监督信息**。现有的实验仅利用了大规模文本，以文本的结构和序列作为监督信息。根据语言学的研究成果，结构和序列信息对于语义层次的学习是不充分的，从结果来看，语言模型对于词之间的语义关系有较好的学习能力，但对于句子、段落或者篇章的表示和学习仍然没有太大的突破。对此，可以通过加入语法依存关系、知识图谱的链接等语义信息，使语言模型更为完善。

（2）**有必要进一步提升模型的解释性**。众所周知，模型越复杂，解释性就会越差。模型的假设通常建立于直觉之上，最终通过实验效果检验。深度学习语言模型的建模以及工程实践中存在许多无法解释的诀窍，因此许多研究仍然缺乏对模型的合理解释，很难说清楚某项具体的优化技术对语言模型的实质性价值。

（3）**增强字符信息在输出端的价值**。本文提出的架构在输入端有效的处理了字符和词项的关系，但是在输出端，仍然忽略了字符的作用。词由字符组成，那么在输出端有必要提供一个字符的解码器，用于预测字符序列。该监督过程会加强模型参数的学习，同时能够解决未登录词的问题。但对于解码器的设计及其融入语言模型架构的方式与途径都需要进一步研讨。

参 考 文 献

[1] 文娟. 统计语言模型的研究与应用[D]. 北京: 北京邮电大学, 2010.

[2] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. Journal of Machine Learning Research, 2003, 3: 1137-1155.

[3] Baroni M, Dinu G, Kruszewski G. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors[C]// Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2014: 238-247.

[4] Bengio S, Bengio Y. Taking on the curse of dimensionality in joint distributions using neural networks[J]. IEEE Transactions on Neural Networks, 2000, 11(3): 550-557.

[5] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[EB/OL]. [2013-09-07]. https://arxiv.org/abs/1301.3781.

[6] Hinton G E. Learning distributed representations of concepts[C]// Proceedings of the Eighth Annual Conference of the Cognitive Science Society, Amherst, 1986, 1: 12.

[7] Le Q, Mikolov T. Distributed representations of sentences and

documents[C]// Proceedings of the 31st International Conference on Machine Learning. 2014, 14: 1188-1196.

[8] Yu M, Dredze M. Improving lexical embeddings with semantic knowledge[C]// Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics,. 2014: 545-550.

[9] Ma W C, Suel T. Structural sentence similarity estimation for short texts[C]// Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference. Association for the Advancement of Artificial Intelligence, 2016: 232-237.

[10] Pennington J, Socher R, Manning C D. GloVe: global vectors for word representation[C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 2014, 14: 1532-1543.

[11] Cohen J D, Servan-Schreiber D, McClelland J L. A parallel distributed processing approach to automaticity[J]. The American Journal of Psychology, 1992, `105(2): 239-269.

[12] Elman J L. Finding structure in time[J]. Cognitive Science, 1990, 14(2): 179-211.

[13] Graves A. Supervised sequence labelling with recurrent neural networks[M]. Berlin: Springer, 2012: 15-35.

[14] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model[C]// Proceedings of the 11th Annual Conference of the International Speech Communication Association, Makuhar, 2010, 2: 3.

[15] Mikolov T, Deoras A, Kombrink S, et al. Empirical evaluation and combination of advanced language modeling techniques[C]// Proceedings of the Twelfth Annual Conference of the International Speech Communication Association, 2011.

[16] 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.

[17] Hochreiter S, Bengio Y, Frasconi P, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies[EB/OL]. [2014-11-19]. https://www.researchgate.net/publication/2839938_Gradient_Flow_in_Recurrent_Nets_the_Difficulty_of_Learning_Long-Term_Dependencies.

[18] Lipton Z C, Berkowitz J, Elkan C. A critical review of recurrent neural networks for sequence learning[EB/OL]. [2015-06-05]. https://arxiv.org/pdf/1506.00019.

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

[20] Kang M, Ng T, Nguyen L. Mandarin word-character hybrid-input Neural Network Language Model[C]// Proceedings of the Conference on the 12th Annual Conference of the International Speech Communication Association, Florence, Italy, 2011: 625-628.

[21] Kombrink S, Mikolov T, Karafiát M, et al. Recurrent neural network based language modeling in meeting recognition[C]// Proceedings of the 12th Annual Conference of the International Speech Communication Association, Florence, Italy, 2011: 2877-2880.

[22] Mikolov T, Kombrink S, Burget L, et al. Extensions of recurrent neural network language model[C]// Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Prague, Czech Republic, 2011: 5528-5531.

[23] Shi Y Z, Zhang W Q, Liu J, et al. RNN language model with word clustering and class-based output layer[J]. EURASIP Journal on Audio, Speech, and Music Processing, 2013: 22.

[24] Karpathy A, Johnson J, Li F F. Visualizing and understanding recurrent networks[EB/OL]. [2015-11-17]. https://arxiv.org/pdf/1506.02078.

[25] Ballesteros M, Dyer C, Smith N A. Improved transition-based parsing by modeling characters instead of words with LSTMs [EB/OL]. [2015-08-11]. https://arxiv.org/abs/1508.00657.

[26] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[EB/OL]. [2014-09-03]. https://arxiv.org/abs/1406.1078.

[27] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]// Proceedings of the Conference on Advances in Neural Information Processing Systems. 2014: 3104-3112.

[28] Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences[EB/OL]. [2014-04-08]. https://arxiv.org/abs/1404.2188.

[29] Kim Y. Convolutional neural networks for sentence classification[EB/OL]. [2014-09-03]. https://arxiv.org/abs/1408.5882.

[30] Tang D Y, Qin B, Liu T. Document modeling with gated recurrent neural network for sentiment classification[C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 2015: 1422-1432.

[31] Wang L, Luís T, Marujo L, et al. Finding function in form: Compositional character models for open vocabulary word representation[EB/OL]. [2016-05-23]. https://arxiv.org/abs/1508.02096.

[32] Kang M, Ng T, Nguyen L. Mandarin word-character hy-

brid-input Neural Network Language Model[C]// Proceedings of the 12th Annual Conference of the International Speech Communication Association, Florence, Italy, 2011: 625-628.

[33] Dos Santos C N, Zadrozny B. Learning character-level representations for part-of-speech tagging[C]// Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014, 32: 1818-1826.

[34] Bojanowski P, Joulin A, Mikolov T. Alternative structures for character-level RNNs[EB/OL]. [2015-11-24]. https://arxiv.org/abs/1511.06303.

[35] Luong M T, Manning C D. Achieving open vocabulary neural machine translation with hybrid word-character models[EB/OL]. [2016-06-23]. https://arxiv.org/pdf/1604.00788.pdf.

[36] 朱德熙. 语法讲义[M]. 北京: 商务印书馆, 1982.

[37] Abadi M, Agarwal A, Barham P, et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems[EB/OL]. [2016-03-16]. https://arxiv.org/abs/1603.04467.

[38] Looks M, Herreshoff M, Hutchins D L, et al. Deep learning with dynamic computation graphs[EB/OL]. [2017-02-22]. https://arxiv.org/abs/1702.02181.

[39] Sak H, Senior A, Beaufays F. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition[EB/OL]. [2014-02-05]. https://arxiv.org/abs/1402.1128.

[40] Bengio Y, Senécal J S. Adaptive importance sampling to accelerate training of a neural probabilistic language model[J]. IEEE Transactions on Neural Networks, 2008, 19(4): 713-722.

[41] Gutmann M, Hyvärinen A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models[C]// Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, 2010: 297-304.

[42] Mnih A, Teh Y W. A fast and simple algorithm for training neural probabilistic language models[EB/OL]. [2012-06-27]. https://arxiv.org/abs/1206.6426.

[43] Dyer C. Notes on noise contrastive estimation and negative sampling[EB/OL]. [2014-10-30]. https://arxiv.org/abs/1410.8251.

[44] Chelba C, Mikolov T, Schuster M, et al. One billion word benchmark for measuring progress in statistical language modeling[EB/OL]. [2014-03-04]. https://arxiv.org/abs/1312.3005.

[45] Ji S H, Vishwanathan S V N, Satish N, et al. BlackOut: Speeding up recurrent neural network language models with very large vocabularies[EB/OL]. [2016-03-31]. https://arxiv.org/abs/1511.06909.

[46] Williams W, Prasad N, Mrva D, et al. Scaling recurrent neural network language models[C]// Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Brisbane, QLD, Australia, 2015: 5391-5395.

[47] Jozefowicz R, Vinyals O, Schuster M, et al. Exploring the limits of language modeling[EB/OL]. [2016-02-11]. https://arxiv.org/abs/1602.02410.

[48] Li X, Qin T, Yang J, et al. LightRNN: Memory and computation-efficient recurrent neural networks[C]// Proceedings of the 30th Conference on Neural Information Processing Systems, Barcelona, Spain, 2016: 4385-4393.

[49] Dauphin Y N, Fan A, Auli M, et al. Language modeling with gated convolutional networks[EB/OL]. [2016-11-23]. https://arxiv.org/abs/1612.08083.

（责任编辑　魏瑞斌）