研究与开发

融合递增词汇选择的深度学习中文输入法

任华健，郝秀兰，徐稳静

（湖州师范学院浙江省现代农业资源智慧管理与应用研究重点实验室，浙江 湖州 313000）

摘　要：输入法的核心任务是将用户输入的按键序列转化为汉字序列。应用深度学习算法的输入法在学习长距离依赖和解决数据稀疏问题方面存在优势，然而现有方法仍存在两方面问题，一是采用的拼音切分与转换分离的结构导致了误差传播，二是模型复杂难以满足输入法对实时性的需求。针对上述不足提出了一种融合了递增词汇选择算法的深度学习的输入法模型并对比了多种 softmax 优化方法。在人民日报数据和中文维基百科数据上进行的实验表明，该模型的转换准确率相较当前最高性能提升了 15%，融合递增词汇选择算法使模型在不损失转换精度的同时速度提升了 130 倍。

关键词：中文输入法；长短期记忆；词汇选择

中图分类号：TP391

文献标志码：A

doi: 10.11959/j.issn.1000–0801.2022294

Deep learning Chinese input method with incremental vocabulary selection

REN Huajian, HAO Xiulan, XU Wenjing

Zhejiang Province Key Laboratory of Smart Management and Application of Modern Agricultural Resources, Huzhou University, Huzhou 313000, China

Abstract: The core task of an input method is to convert the keystroke sequences typed by users into Chinese character sequences. Input methods applying deep learning methods have advantages in learning long-range dependencies and solving data sparsity problems. However, the existing methods still have two shortcomings: the separation structure of pinyin slicing in conversion leads to error propagation, and the model is complicated to meet the demand for real-time performance of the input method. A deep-learning input method model incorporating incremental word selection methods was proposed to address these shortcomings. Various softmax optimization methods were compared. Experiments on People's Daily data and Chinese Wikipedia data show that the model improves the conversion accuracy by 15% compared with the current state-of-the-art model, and the incremental vocabulary selection method makes the model 130 times faster without losing conversion accuracy.

Key words: Chinese input method, long short-term memory, vocabulary selection

---

收稿日期：2022–07–08；**修回日期**：2022–12–07

通信作者：郝秀兰，hxl2221_cn@zjhu.edu.cn

基金项目：浙江省现代农业资源智慧管理与应用研究重点实验室基金项目（No.2020E10017）

Foundation Item: The Foundation of Zhejiang Province Key Laboratory of Smart Management and Application of Modern Agricultural Resources (No.2020E10017)

0　引言

英语等拉丁语系用户可以通过计算机键盘直接输入想要的词句，然而由于汉字总量十分庞大，直接输入并不现实，中文用户需要借助输入法进行汉字输入。输入法的核心任务是将用户输入的按键序列转化为相应的汉字序列。理想情况下，输入法的转换结果应与用户期望输入的汉字序列相同。输入法的转换准确率和转换速度直接影响着输入效率和用户体验。

输入法内核的转换任务与语音识别、机器翻译相似，可以看作一项序列解码任务。传统的统计机器学习算法采用 n 元语法语言模型[1]计算词网格中最优路径的概率。该模型需要通过增大 n 值学习语料中的长距离特征，但过大的 n 值会带来数据稀疏问题，在实践中普遍采用三元语法语言模型。相较于传统模型，基于深度学习算法的语言模型具有能够在学习长距离特征的同时不易受数据稀疏问题制约的特点，因此获得了广泛研究。在机器翻译[2]、语音识别[3]、情感分析[4-5]等自然语言处理任务中，基于深度学习算法的模型取得了更高的性能。近几年来张量处理单元[6]（tensor processing unit，TPU）等硬件方面的研究进展为深度学习模型的应用提供了更加广阔的前景。

然而基于深度学习的模型仍然难以直接应用在输入法的转换任务中。与语音识别和机器翻译多由云服务器提供不同，输入法需要运行在多样化的用户设备上并提供实时交互服务。即使有很多输入法厂商提供了云候选，但是其转换结果因为时延较高，往往只能作为次后选。一个能够适用于输入法转换任务的深度学习模型需要满足以下两点要求：能够为每次用户的按键输入提供低时延的转换结果；能够在转换完成后允许用户对候选结果进行编辑。

首先，输入法转换任务的输入序列是逐个获得的，并且需要在每一次收到按键信息后立即返回词网格中的最优路径。现存的优化手段[7-8]普遍着眼于提升对序列进行批处理时的速度。文献[9]的方法融合了前缀树，但是无法保证在最坏情况下运算时间仍然能够满足实时应用的需求。其次，输入法在给出转换结果后，允许用户以手动选择候选词的方式对词网格进行编辑。这一特点限制了多种端到端模型[10-11]的使用，因为它们都没有提供允许用户编辑部分转换结果的方法。

为满足输入法转换任务的要求，本文提出了一个新的基于深度学习算法的输入法模型。在该模型中，用户输入首先被切分成拼音音节，通过维特比解码器构造出相应的词网格。然后通过融合递增词汇选择的长短期记忆[12]（long short-term memory，LSTM）语言模型计算最优路径的概率。该语言模型在解码过程的第 t 步构建一个输入法总词汇表的子集 V_t。由于 V_t 中词汇的数量通常在总词汇量的 1%以下，仅在 V_t 上执行 softmax 运算，模型消耗的时间相较基线模型有了大幅降低。

1　相关工作

早期相关研究常采用基于统计语言模型[13]或统计机器翻译[14]的方法。近年来出现了采用深度学习算法的研究。文献[15]将输入法转换任务看作机器翻译，使用了一个融合注意力机制的编码器－解码器模型完成音字转换任务，并使用信息检索技术对模型的联想输入功能进行了增强。文献[16]在基于 LSTM 的编码器－解码器模型基础上，通过在线词汇更新算法扩充输入法词汇表，提高了转换准确率。文献[17]通过在输入法中增加输入上下文的编码模块，提高了模型在简拼模式下的表现。文献[18]使用一个字符级中文预训练语言模型作为输入法核心完成音字转换，并获得了当前最高性能（state-of-the-art，SOTA），同时研究了基于预训练语言模型的输入法在简拼场景下的表现。现有模型普遍比较复杂，需要在 GPU 的支持下才能运行，文献[15]则是将输入法转换任务和联想功

能等功能全部架设在云服务器上。本文工作的重点是在计算资源有限的用户设备上实现基于神经网络的实时输入法。

具有大量词汇的 softmax 层是神经网络语言模型中计算量最大的操作，目前针对 softmax 瓶颈的解决方法主要有两种。其一是对 softmax 运算进行近似。文献[19]提出的差异化 softmax（differentiated softmax）及文献[20]提出的自适应 softmax（adaptive softmax）将语言模型中频率较低的长尾词汇按词频进行分割，并减少其嵌入向量维度，从而降低 softmax 运算的计算量。对于预测任务，其前 k 个候选的重要性较高，文献[21]提出的奇异值分解 softmax（singular value decomposition softmax，SVD-softmax）对权重矩阵进行奇异值分解，并使用得到的低秩矩阵进行 softmax 运算。在解码过程中，如果目标词汇有一个确定的范围，则可以使用结构化输出层[22]避免计算所有词汇的概率分布。文献[23]提出了一种快速词图解码方法，该方法可以在对数时间复杂度内找到神经网络语言模型前 k 个最有可能的预测。

另一种解决 softmax 瓶颈的方法是对推理过程的词汇进行操作，与机器翻译任务中的方法相似。即先针对待翻译的句子构建一个可能使用的总词汇表的子集，然后在这一子集上进行 softmax 运算[24-25]。文献[26]提出了一种针对日语输入法的词汇选择方法，降低了模型推理时间。基于单字的模型，如单字循环神经网络[27-28]（recurrent neural network，RNN）语言模型相较于基于词语的模型可以大幅减少模型中的词汇量，然而汉字的数量仍然不是一个小数目，并且这种方法无法利用词语的信息，所以会导致模型准确率大幅下降。

受上述研究启发，本文提出一种融合递增词汇选择方法的深度学习输入法模型，该模型使用词网格进行解码，可以在用户输入过程中递增地构建所需词汇表，提升了转换准确率并降低了转换任务中序列解码所需的时间。

2 模型建立

2.1 深度学习输入法模型

本文提出的长短期记忆语言模型输入法模型总体架构如图 1 所示。考虑模型的实时性目标，本文采用一个 LSTM 语言模型计算词网格解码器中不同句子的路径概率。由于序列到序列[10]等模型不能动态递增地生成转换结果，本文没有采用这类模型。

在音字转换任务中，用户按键输入首先被切分为拼音音节序列。对第 t 步输入产生的音节序列$(x_1,\cdots,x_t)$，用式（1）查找字典中满足该序列所有后缀的词语组成词网格。

$$D_t=\bigcup_{i=1}^{t}\text{fetch}\left(x_i,\cdots,x_t\right) \tag{1}$$

其中，fetch(·)函数从输入法总词汇表中查找并返回所有符合部分序列$(x_i,\cdots,x_t)$的词汇。例如，对于音节序列“zai hu zhou”，产生的子词汇集 D_t 包含

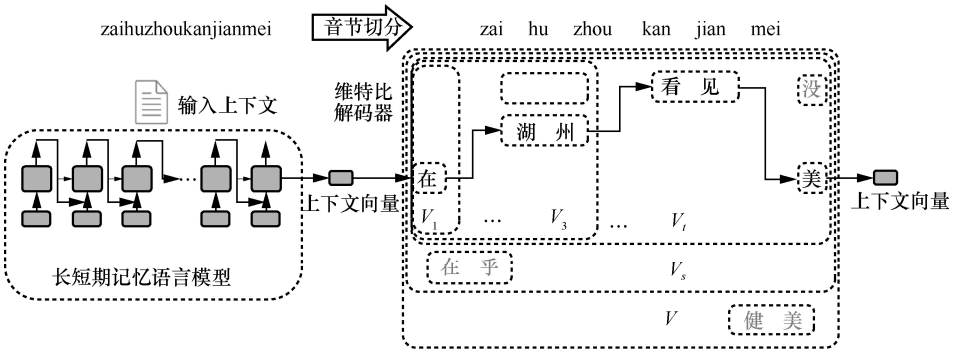

图 1 长短期记忆语言模型输入法模型总体架构

所有匹配“zai hu zhou”“hu zhou”和“zhou”的结果。

对于词汇表子集中的任意词语 $w_t \in D_t$，为了构造一个以 w_t 结尾的候选 π_t，之前步骤生成的部分候选 π_{t-l} 被用作计算 $p(w_t|\pi_{t-l})$ 的上下文，其中，l 是 w_t 的长度，只有字符数量匹配的部分候选才会被连接成一个完整的候选。最后，使用一个束宽度为 B 的维特比解码器对候选进行排序。

在维特比解码器中，使用 LSTM 语言模型计算 $p(w_t|\pi_{t-l})$。对于每一个输入序列，音字转换任务需要进行 $B\times|V|\times T$ 步 LSTM 运算，其中 $|V|$ 是词汇表中词汇的数量，T 是音节序列的长度，输入法主要的计算量集中于 LSTM 单元和 softmax 的矩阵运算。假设一个输入法的词汇总量为 10 万，使用一个嵌入向量维度和隐藏维度均为 512 的 LSTM 语言模型，以矩阵中的乘法为基准操作估计。LSTM 单元中有 4 处矩阵乘法，每处为两个 512 维向量与 512 维权重方阵相乘，为操作数量的高阶项，总计 512×512×2×4 次乘法操作，数量约为 200 万。在 softmax 层，需要将 512 维向量投影成 10 万维以计算每个词汇的概率，为操作数量的高阶项，总计 512×100 000 次乘法操作，数量约为 5 000 万，占比为 51 200 000/53 200 000，为总量的 96%以上。对于应用于中文输入法的拥有大量词汇的语言模型而言，其时间效率瓶颈在于 softmax 操作中的大量矩阵运算。

2.2　递增词汇选择模型

为了降低解码过程中 softmax 的运算量，本文使用递增词汇选择算法改进了 LSTM 基线模型，如算法 1 所示。

算法 1　递增词汇选择

初始化：束宽度 B，候选字典 $\Pi[t]$，采样词汇 V_s，当前时间步 t

V_t = 式（2）中的词汇子集

$V_t = V_t \cup V_s$

V_{fix} = 空集

for $k = t-1$ to 1 do

$V_{\text{fix}} = V_{\text{fix}} \cup (V_t \setminus V_k)$

$V_k = V_t$

重新计算 $\Pi[t]$

$\Pi[t] = \Pi[t]$ 中最优的 B 个候选

输出 $\Pi[t]$

其中，Π 是存储每一步最优候选及其 LSTM 隐藏状态的字典。定义 $\Pi[\cdot]$ 为某一特定时间步用于获取当前最优候选的查找操作。Π 的初始内容为上一次音字转换任务结果及其 LSTM 隐藏状态，第一次转换任务时为空。

在第 t 步，接收到一个新的拼音音节 x_t 时，算法构造一个相应的词汇表 V_t，如式（2）所示，V_t 中覆盖了所有到第 t 步为止可能出现的词语。

$$V_t = \bigcup_{i=1}^{t} D_i \tag{2}$$

对候选 (π_{t-l}, w_t) 进行排序需要计算概率 $p(w_t|\pi_{t-l})$，而该概率已经在 $\Pi[t-l]$ 计算过并存储在字典中了。然而，因为 V_t 的构建是递增的，所以在之前步骤计算的概率分布中可能并没有包含 w_t。所以，为了计算 $\Pi[t]$，需要修复之前概率分布计算中缺失的词汇。可以通过式（3）修正第 k 步的概率。

$$P(y_k = i|\boldsymbol{h}_k) = \frac{\exp(\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{W}_i)}{\sum_{j\in V_k}\exp(\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{W}_i) + \sum_{j\in d}\exp(\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{W}_j)} \tag{3}$$

其中，$\boldsymbol{W}$ 是输出层的权重矩阵，$\boldsymbol{h}_k$ 是对应候选 π_k 的 LSTM 隐藏状态，d 是词汇集 V_t 和 V_k 的差集。在当前第 k 步向词网格中新增 d 中的词汇后，导致与之前步骤进行 softmax 运算时的分母不一致，所以该步在之前 softmax 结果的分母中加入新增词汇对应的输出单元的值，即分母中的第二项，可以保持各步涉及词汇进行 softmax 概率计算时的分母相同。因为词网格解码器中的每一条路径都有不同的缺失词汇，所以先取这些词汇的并集

V_{fix}，再重新计算所有路径的概率。因为在一次输入前几步的词汇集比较小，所以引入从整体词汇表中采样出的一个特定子集 V_s。后续实验中对比了两种不同采样方式的效果：其一是顶端采样，即从总词汇表中选出词频最高的若干词作为 V_s；其二是均匀采样，即从总词汇表中随机选出若干词汇作为 V_s。当仅使用 V_t 时相当于去掉了 softmax 分母中的部分项，此时会导致概率偏大，将采样出的词汇集与之合并，同时使用 V_t 和 V_s，使词语的概率更加接近原始概率。计算结束后，$\Pi[t]$ 中最优的 B 个候选将会作为输入法音字转换任务的结果返回给用户。

3 实验与分析

3.1 数据集和评价标准

为了验证模型的效果，本文分别在两个不同的中文数据集上进行了实验。

（1）人民日报实体数据集。该数据集语料提取自 1992—1998 年的人民日报新闻内容，是中文输入法研究中常用的数据集[14]。其中，训练集包含了 504 万个最大输入单元，测试集包含 2 000 个最大输入单元。最大输入单元的定义为句子中最长连续汉字序列[15]。

（2）中文维基百科数据集。由于人民日报数据集语料内容较为陈旧，数据量也相对较小，本文选择在中文维基百科数据集上完成模型推理时间的实验。该数据集语料为截至 2022 年 1 月 1 日维基百科所有中文词条内容。原始语料为 9.58 GB，预处理时先将句子切分成最大输入单元后再进行分词和标注拼音形成汉字－拼音平行语料。数据按照 7:2:1 的比例划分为训练集、验证集和测试集，其中训练集包含 6 660 万个最大输入单元，验证集包含 1 902 万个最大输入单元，测试集包含 951 万个最大输入单元。

本文使用了输入法研究中[15-18]普遍采用的“Top k Accuracy”作为评价指标，这一指标表示在输入法提供的前 k 个候选中用户期待输入的汉字序列的命中率，其计算式如下。

$$\text{Top } k \text{ Accuracy} = \frac{\sum_{i=1}^{N} \text{exist}(k)}{N} \tag{4}$$

其中，exist(·)为判断前 k 个候选中是否存在期待输入汉字序列的函数，当前 k 个候选中存在期待的候选时返回值为 1，否则为 0；N 为测试集数量。

3.2 实验设置

实验中使用 TensorFlow 实现模型，训练使用的 GPU 型号为 NVIDIA Tesla V-100。为防止模型过拟合，引入了丢弃机制，参数值为 0.9。模型使用 Adam 优化器，学习率固定为 0.001。对于所有实验，这些参数都是相同的。

在人民日报数据集上进行实验时，选择频率最高的前 5 万词作为输入法词汇表，与文献[16]中的对比模型保持一致，这些词汇覆盖了语料内容的 99%。词汇表中的词汇按照频率排列，高频词在前。由于词汇量小，训练时批处理大小设置为 2 048。

在中文维基百科数据集上进行实验时，先使用 jieba 分词工具对语料进行分词和标注拼音。参考常见的开源拼音输入法的词汇数量，sunpinyin 约为 8.4 万、libpinyin 约为 11.1 万、rime 约为 9.9 万。同时，自归一化模型在不同词汇量下的表现曲线如图 2 所示，更大的词汇量可以带来更高的准确率，但是其边际收益是递减的。

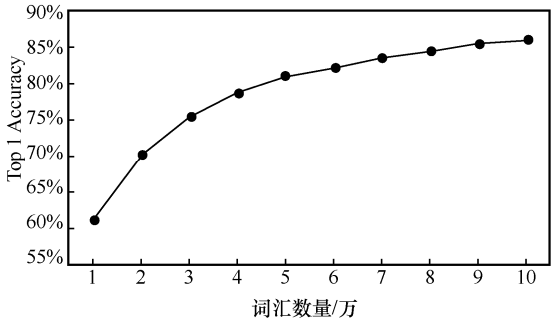

图 2 自归一化模型在不同词汇量下的表现曲线

实验中依照上述经验值，选择频率最高的前 10 万词作为输入法词汇表，这些词汇覆盖了语料内容的 96%，词汇数量作为受控变量在不同各实验中保持一致。文献[18]使用分类语料对按照领域信息划分词汇表的方法进行了研究，本文计划将这一课题作为未来的研究方向之一。词汇表同样按照词频降序排列。训练批处理大小为 1 024。由于划分了验证集，所以在训练中使用了早停策略防止过拟合，当验证集上的语言模型困惑度连续两轮不再下降时将提前停止训练。困惑度计算式如下。

$$\mathrm{PP}(S)=P\left(w_1 w_2 \cdots w_n\right)^{-\frac{1}{N}}=\sqrt[N]{\prod_{i=1}^{N}\frac{1}{P\left(w_i \mid w_1 \cdots w_{i-1}\right)}} \tag{5}$$

其中，PP 是困惑度，S 是语料，w_i 是其中第 i 个词语，N 是语料中词语总数。困惑度能够反映语言模型对数据的拟合能力，一般困惑度低的模型能够在推理任务中获得较高的准确率。训练模型时使用困惑度的原因在于困惑度是可微的，而准确率是不可微的。在对比模型表现时使用准确率这一指标可以使结果更加直观。

在 GPU 上完成训练后，将模型参数导出至文件，供词网格解码器在解码过程中使用。输入法的解码实验在 Intel(R) Xeon(R) E5-2650 v3 上完成，模型使用 numpy 实现，其结构与 TensorFlow 模型相同。

3.3　转换准确率

输入法转换准确率对比结果见表 1，实验将本文模型与如下基线模型进行了对比。

（1）三元语法语言模型输入法。该模型为开源输入法中广泛使用的语言模型，使用开源工具包斯坦福研究院语言建模工具包[1]（Stanford research institute language modeling toolkit，SRILM）进行训练，平滑方法使用改良的 Kneser-Ney 算法[29-30]。为了获得更高的转换准确率，没有设置截止频率或对模型进行剪枝。

（2）谷歌拼音输入法。谷歌拼音输入法是一个提供了调试接口的商业输入法。

（3）在线词汇更新音字转换（online updated vocabulary pinyin to character，Online P2C）。该模型为文献[16]在 2019 年提出的方法。模型使用一个带有注意力机制的编码器—解码器结构完成转换任务，并使用在线词汇更新算法对其进行了增强。

（4）生成式预训练音字转换（generative pre-training pinyin to character，GPT P2C）。该模型为文献[18]在 2022 年提出的方法。模型使用一个字符级 GPT 输入 800 GB 语料在 32 个 NVIDIA Tesla V-100 上训练完成，并且使用拼音嵌入对模型进行了增强。

表 1　输入法转换准确率对比结果

模型	Top 1 Accuracy	Top 10 Accuracy
三元语法	54.4%	66.4%
谷歌拼音输入法	70.9%	82.3%
Online P2C	71.3%	81.3%
GPT P2C	73.2%	85.5%
LSTM 基线	88.6%	97.6%

本文 LSTM 模型词向量嵌入维度和隐藏维度均为 512，并使用文献[31]中的方法对输入和输出的词向量进行了绑定，这一方法可以节约存储空间且几乎不造成精度损失。

由实验结果可知，本文模型在 Top 1 Accuracy 上相比 GPT P2C 提升了 15.4%，取得了新的最高性能。这表明了本文方法在提升转换准确率方面的有效性。

3.4　模型运行时间

本节使用上一节 LSTM 模型作为基线，对比了多种加速方法的效果。本文仅计算了参与语言模型概率运算的组件的执行速度，其中包含 LSTM 层和 softmax 层的运算时间。实验中，在

LSTM 单元和 softmax 单元前后分别插入计时点记录单次运算时间，所有推理任务结束后求平均值作为单次 LSTM 和 softmax 运算的运行时间，单步时间为两者相加之和。构造词网格的组件等则没有被包含进来，一方面因为这些组件对于不同模型是共用的，另一方面它们的具体实现对运行时间影响很大，不具有参考价值。另外，文献[16]和文献[18]中的两个深度学习模型由于复杂度高，需要 GPU 支持运算，强行转移到 CPU 上进行推理任务带来的时间消耗是实际应用场景下无法承受的，因此没有参与对比。

单步转换时延对于需要满足用户实时输入需求的输入法而言是十分重要的，表 2 对比了每一步收到按键信息之后的平均解码时间。解码整个输入拼音序列的时间消耗与解码步数（拼音字母数量）成正比。模型运行时间对比见表 2，同时单独提供了 softmax 的运算时间以供对比。

由表 2 可以看出，本文提出的融合递增词汇选择模型的 softmax 运算速度约是 LSTM 基线模型的 130 倍。输入法的整个词汇表中含有 10 万个词汇，而实验中词网格解码器使用到的词汇只有几百个，对上述实验词网格解码器中的词汇数量进行统计，仅有 188 个句子的词汇数量超过了总词汇量的 1%，约占实验中所有句子的 2%，其中最多的一句词网格包含了 2 302 词。递增词汇选择模型只需要 3.2 ms 处理一个新加入序列的按键信息。这样的解码速度可以在计算资源有限的用户终端设备上满足实时交互需求。

表 2 中还对比了递增词汇选择的各种采样方法。可以看出，对总词汇集进行采样可以缩小与基线模型之间的准确率差距。在实验中，顶部采样和均匀采样的词汇数量均设置为 500 词。

另外，还可以使用自归一化[32]作为 softmax 近似。原始 softmax 的对数概率计算式为：

$$\ln(P(x)) = \ln\left(\frac{e^{U_r(x)}}{Z(x)}\right) = U_r(x) - \ln(Z(x)) \quad (6)$$

其中，

$$Z(x) = \Sigma_{r'=1}^{|V|} e^{U_{r'}(x)} \quad (7)$$

其中，x 是词汇，U 是输出单元的值，r 是当前词汇序号。自归一化通过显式地令分母尽量接近 1 简化 softmax 运算。因此需要在训练过程的目标函数中加入额外的归一化项，如式（8）所示。

$$L = \sum_i \left[\ln\left(P\left(x_i\right)\right) - \alpha\left(\ln\left(Z\left(x_i\right)\right) - 0\right)^2 \right] = \sum_i \left[\ln\left(P\left(x_i\right)\right) - \alpha \ln^2\left(Z\left(x_i\right)\right) \right] \quad (8)$$

其中，α 是自归一化权重，实验中设置为 0.1。在使用自归一化时，为了保证网络在初始时就是自归一的，输出层的初始偏置设置为 $\ln(1/|V|)$。在解码过程中，直接使用 $U_r(x)$代替 $\ln(P(x_i))$。

差异化 softmax 和自适应 softmax 可以降低超过一半的 softmax 运算时间，然而由于输入法词汇量巨大，这些方法对速度提升有限。

对于单字模型，实验使用了一个嵌入向量

表 2　模型运行时间对比

模型	单步时长/ms	softmax 时长/ms	Top 1 Accuracy	Top 10 Accuracy
LSTM 基线	172.1	169.8	86.4%	97.1%
单字模型	6.8	4.6	66.8%	79.5%
单字模型束宽度 50	14.2	9.3	69.1%	86.3%
差异化 softmax	43.0	39.3	84.7%	96.2%
自适应 softmax	41.4	39.0	84.4%	96.1%
递增词汇选择	3.2	1.3	84.6%	95.8%
+顶部采样	4.0	2.0	86.1%	96.8%
+均匀采样	4.0	2.0	84.7%	95.8%
+自归一化	3.2	1.2	86.1%	96.9%

维度和隐藏状态向量维度均为 512 的 LSTM 语言模型。该模型将输入法词汇量从 10 万降低到了 5 000，然而这一数量仍然会带来不小的开销。在转换任务中，单字模型的准确率损失较大，需要采用更大的束宽度提高转换准确率，实验中大的束宽度设置为 50。

递增词汇选择算法中用于修正词汇表的时间由于依赖具体实现没有给出，而且在实际应用场景中，用户并不会一次性输入一个完整的句子而是分成多个片段输入，这些时间与 softmax 运算消耗的时间相比是很小的。

4　结束语

本文提出了一种融合递增词汇选择的深度学习输入法模型。相较于传统 n 元语法模型和其他深度学习输入法，该模型能够提供更高的转换准确率；相较于未作改进的基线模型，该模型能够在不损失转换准确率的同时缩短 softmax 运算所需的时间。本文在真实中文数据集上进行了实验，实验结果表明，该模型能够满足输入法的实时交互需求。后续笔者将在其他数据集和其他中文输入方法上对本文模型做进一步测试和增强。

参考文献：

[1] STOLCKE A. SRILM - an extensible language modeling toolkit[C]//Proceedings of 7th International Conference on Spoken Language Processing (ICSLP 2002). ISCA: ISCA, 2002: 901-904.

[2] XIAO Y L, LIU L M, HUANG G P, et al. BiTIIMT: a bilingual text-infilling method for interactive machine translation[C]//Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2022: 1958-1969.

[3] PARASKEVOPOULOS G, PARTHASARATHY S, KHARE A, et al. Multimodal and multiresolution speech recognition with transformers[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2020: 2381-2387.

[4] DING Z X, XIA R, YU J F. End-to-end emotion-cause pair extraction based on sliding window multi-label learning[C]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg, PA, USA: Association for Computational Linguistics, 2020: 3574-3583.

[5] DING Z X, XIA R, YU J F. ECPE-2D: emotion-cause pair extraction based on joint two-dimensional representation, interaction and prediction[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2020: 3161-3170.

[6] PANDEY P, BASU P, CHAKRABORTY K, et al. GreenTPU: improving timing error resilience of a near-threshold tensor processing unit[C]//Proceedings of DAC '19: Proceedings of the 56th Annual Design Automation Conference. [S.l:s.n.], 2019. 2019: 1-6.

[7] LIU X Y, CHEN X, WANG Y Q, et al. Two efficient lattice rescoring methods using recurrent neural network language models[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2016, 24(8): 1438-1449.

[8] LEE K, PARK C, KIM N, et al. Accelerating recurrent neural network language model based online speech recognition system[C]//Proceedings of 2018 IEEE International Conference on Acoustics, Speech and Signal Processing. Piscataway: IEEE Press, 2018: 5904-5908.

[9] CHEN X, LIU X Y, WANG Y Q, et al. Efficient training and evaluation of recurrent neural network language models for automatic speech recognition[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2016, 24(11): 2146-2157.

[10] CHO K, VAN MERRIENBOER B, GULCEHRE C, et al. Learning phrase representations using RNN encoder–decoder for statistical machine translation[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg, PA, USA: Association for Computational Linguistics, 2014: 1724-1734.

[11] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017, 22(7): 139-147.

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

[13] CHEN Z, LEE K F. A new statistical approach to Chinese Pinyin input[C]//Proceedings of the 38th Annual Meeting on Association for Computational Linguistics - ACL '00. Morristown, NJ, USA: Association for Computational Linguistics, 2000: 241-247.

[14] YANG S, ZHAO H, LU B. A machine translation approach for Chinese whole-sentence Pinyin-to-character conversion[C]//Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation. [S.l.:s.n.], 2012: 333-342.

[15] HUANG Y F, LI Z C, ZHANG Z S, et al. Moon IME: neural-based Chinese pinyin aided input method with customizable association[C]//Proceedings of ACL 2018, System Demonstrations. Stroudsburg, PA, USA: Association for Computational Linguistics, 2018: 140-145.

[16] ZHANG Z S, HUANG Y F, ZHAO H. Open vocabulary learning for neural Chinese pinyin IME[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019: 1584-1594.
[17] HUANG Y F, ZHAO H. Chinese pinyin aided IME, input what You have not keystroked yet[C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: Association for Computational Linguistics, 2018: 2923-2929.
[18] TAN M H, DAI Y, TANG D Y, et al. Exploring and adapting Chinese GPT to pinyin input method[C]//Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2022: 1899-1909.
[19] CHEN W L, GRANGIER D, AULI M. Strategies for training large vocabulary neural language models[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2016: 1975-1985.
[20] JOULIN A, CISSÉ M, GRANGIER D, et al. Efficient softmax approximation for GPUs[C]//International Conference on Machine Learning. [S.l.:s.n.], 2017: 1302-1310.
[21] SHIM K, LEE M, CHOI I, et al. SVD-softmax: Fast softmax approximation on large vocabulary neural networks[J]. Advances in Neural Information Processing Systems, 2017, 30: 5463-5473.
[22] 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, 2013: 22.
[23] Zhang M J, Wang W H, Liu X D, et al. Navigating with graph representations for fast and scalable decoding of neural language models[J]. Advances in Neural Information Processing Systems, 2018, 31: 6308-6319.
[24] MI H T, WANG Z G, ITTYCHERIAH A. Vocabulary manipulation for neural machine translation[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2016: 124-129.
[25] JEAN S, CHO K, MEMISEVIC R, et al. On using very large target vocabulary for neural machine translation[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2015: 1-10.
[26] YAO J, SHU R, LI X, et al. Enabling real-time neural IME with incremental vocabulary selection[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. [S.l.:s.n.], 2019: 1-8.
[27] MIKOLOV T, KARAFIÁT M, BURGET L, et al. Recurrent neural network based language model[C]//Proceedings of Interspeech 2010. ISCA: ISCA, 2010, 2(3): 1045-1048.
[28] ELMAN J L. Finding structure in time[J]. Cognitive Science, 1990, 14(2): 179-211.
[29] KNESER R, NEY H. Improved backing-off for M-gram language modeling[C]//Proceedings of 1995 International Conference on Acoustics, Speech, and Signal Processing. Piscataway: IEEE Press, 1995: 181-184.
[30] JAMES F. Modified kneser-ney smoothing of n-gram models[R]. Research Institute for Advanced Computer Science, 2000.
[31] PRESS O, WOLF L. Using the output embedding to improve language models[C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. Stroudsburg, PA, USA: Association for Computational Linguistics, 2017: 157-163.
[32] DEVLIN J, ZBIB R, HUANG Z Q, et al. Fast and robust neural network joint models for statistical machine translation[C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2014: 1370-1380.

[作者简介]

任华健（1994– ），男，湖州师范学院硕士生，主要研究方向为自然语言处理、中文输入法。

郝秀兰（1970– ），博士，女，湖州师范学院副教授、硕士生导师，主要研究方向为智能信息处理、数据与知识工程、自然语言理解等。

徐稳静（1998– ），女，湖州师范学院硕士生，主要研究方向为自然语言处理、虚假新闻检测。