文章编号:1006-2475(2016)12-0029-05

循环神经网络结构中激活函数的改进

叶小舟,陶飞飞,戚荣志,张云飞,周思琪,刘 璇

(河海大学计算机与信息学院,江苏 南京 210098)

摘要:循环神经网络相比于其他深度学习网络,优势在于可以学习到长时依赖知识,但学习过程中的梯度消失和爆炸问题严重阻碍了知识的按序传播,导致长时依赖知识的学习结果出现偏差。为此,已有研究主要对经典循环神经网络的结构进行改进以解决此类问题。本文分析 2 种类型的激活函数对传统 RNN 和包含门机制 RNN 的影响,在传统 RNN 结构的基础上提出改进后的模型,同时对 LSTM 和 GRU 模型的门机制进行改进。以 PTB 经典文本数据集和 LMRD 情感分类数据集进行实验,结果表明改进后的模型优于传统模型,能够有效提升模型的学习能力。

关键词:深度学习; 循环神经网络; 激活函数; LSTM 模型; GRU 模型

中图分类号:TP311 文献标识码:A doi: 10.3969/j.issn.1006-2475.2016.12.006

Improvement on Activation Functions of Recurrent Neural Network Architectures

YE Xiao-zhou, TAO Fei-fei, QI Rong-zhi, ZHANG Yun-fei, ZHOU Si-qi, LIU Xuan

(College of Computer and Information, Hohai University, Nanjing 210098, China)

Abstract: Recurrent neural network has the advantage of learning long term dependencies, in contrast with other deep learning network architectures. However, the problems of vanishing and exploding gradients seriously obstruct the transmission of information over time, resulting in the deviation of learning long term dependencies. Hence, a great deal of studies focus on the adaption of classical recurrent neural network architectures. In this paper, we analyse the effect of two types of activation function for basic RNN and RNNs with gating mechanism. An improved model based on the basic RNN structure is proposed. The improved gating mechanisms of LSTM model and GRU model are proposed. Experiments on PTB classical dataset LMRD feeling classified dataset show that the improved models are advanced than traditional models and greatly improve the learning ability of the models.

Key words: deep learning; recurrent neural network; activation function; LSTM model; GRU model

0 引 言

人工神经网络是机器学习领域的一个分支。深度学习源于人工神经网路,能更好地模拟大脑结构,实现认知过程逐层抽象,解决深度不足出现的问题[1]。深度学习分支较多,目前的研究热点是卷积神经网络和循环神经网络(Recurrent Neural Networks,RNN)[2]。RNN 是一种学习能力很强的网络系统,能够处理前后关联的信息,适用于处理时间序列数据,例如语音识别、文本生成、机器翻译、序列预测等。为了计算每层网络的误差值,RNN 通常使用时间进化反传算法(Back-Propagation Through Time,BPTT)[3]。但是 BPTT 无法解决长时依赖问题,因此该算法会带来梯度消失和梯度爆炸问题。

为解决梯度消失和梯度爆炸这 2 个问题,在优化学习算法和配置网络的技巧方面,研究人员提出了很多改进方法。在设计和构建新的网络结构方面,Hochreiter 等[4]在 1997 年提出了改进结构 Long Short Term Memory(LSTM)来解决梯度消失的问题。此后,研究人员基于 LSTM 结构进行了大量的探索,比较经典的是 Gers 等[5-6]在 2000 年提出了增加窥视孔连接(peephole connection)和遗忘门的 LSTM 结构。LSTM 的核心思想是门机制,即每个神经元内部由输入门来决定允许有多少信息进入细胞;输出门用来控制输出的信息量;遗忘门控制信息衰减的速率;记忆细胞内存储的神经元信息可以随着时间进行传递。受到

收稿日期:2016-09-09

基金项目:国家科技支撑计划项目(2013BAB06B04; HNKJ13-H17-04); 国家自然科学基金面上资助项目(61272543); 水利部公益性行业科研专项重点项目(201501007); NSFC - 广东联合基金重点项目(U1301252)

作者简介:叶小舟(1992-),男,江苏南京人,河海大学计算机与信息学院硕士研究生,研究方向:数据挖掘; 陶飞飞(1980-),男,硕士生导师,博士,研究方向:大数据技术。

LSTM 的门机制启发，Cho 等[7] 在 2014 年提出了 Gated Recurrent Unit(GRU) 结构。GRU 同 LSTM 一样都包含门机制，但是 GRU 结构合并了部分门，并且神经元内部不包含记忆细胞，从而简化了 LSTM。Chung 等[8] 在 2014 年将 GRU 和 LSTM 与传统的 RNN 进行对比，发现有门机制的 GRU 和 LSTM 优于传统的 RNN，而 GRU 和 LSTM 在不同训练集上则各有千秋。探究门机制为何能够学习到长时依赖知识以及是否有更好的门结构成为研究的热点。Graves 等[9] 在 2013 年将 LSTM 的输入激活函数由 sigmoid 函数改为 tanh 函数，尝试通过修改激活函数来调整门结构。传统的激活函数一般是 sigmoid 函数和 tanh 函数，其特点是输出值在较小的有限区间内。近几年引起关注的激活函数 ReLU[10] 的输出值则在无限区间。2015 年，Le 等[11] 将 ReLU 函数应用于 RNN，发现该模型在某些数据集上取得与 LSTM 相当的效果。同年，Greff 等[12] 将 8 个经典的 LSTM 变种结构进行了比较，发现输出激活函数是 LSTM 门结构中最重要的部分之一，确保了细胞状态的稳定。

softplus 函数图形和 ReLU 函数图形较为相似。本文通过 ReLU 函数和 softplus 函数这一相同类型的函数，来改进拥有门机制的 RNN，对比门结构输出部分中的传统激活函数，探究 2 种类型的激活函数对包含门机制 RNN 的作用。基于 Le 等[11] 人的研究，提出基于 softplus 函数的传统 RNN 模型，同时采用对比方法探究同类型函数，softplus 函数和 ReLU 函数对模型的影响。

1 预备知识

理论上，RNN 能够学习到任意时间长度序列的知识。然而随着间隔增长，伴随梯度消失和梯度爆炸，RNN 变得难以学习到连接之间的关系，产生了长时依赖问题。LSTM 结构能够学习长时依赖知识，其核心思想在于记忆细胞和门机制[4]。如图 1 所示，记忆细胞随着整个链条从头到尾运行，中间只有少量交互，保证了信息流动的顺畅和稳定。门机制是选择性地让信息通过，对神经元进行增加或者删除信息。门机制中的各个门和记忆细胞的表达式如下：

LSTM 遗忘门表达式：

$$f_t = s(W_f \cdot [h_{t-1}, x_t] + b_f)$$

LSTM 输入门表达式：

$$i_t = s(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

LSTM 细胞更新表达式：

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

LSTM 输出门表达式：

$$o_t = s(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(c_t)$$

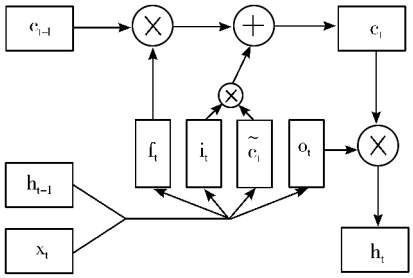

图 1 LSTM 结构示意图

受 LSTM 模型门机制启发，GRU 结构将 LSTM 结构中的输入门和遗忘门结合成一个单独的更新门，合并了记忆细胞和隐含状态，同时也做了一些调整[7]。如图 2 所示，该模型比标准 LSTM 模型简单，得到越来越广泛的使用。GRU 中各个门的表达式如下：

GRU 更新门表达式：

$$z_t = s(W_z \cdot [h_{t-1}, x_t])$$

GRU 重置门表达式：

$$r_t = s(W_r \cdot [h_{t-1}, x_t])$$

GRU 输出部分表达式：

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

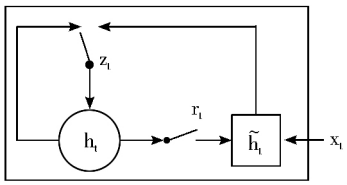

图 2 GRU 结构示意图

ReLU 函数和 softplus 函数的表达式分别为 $f(x) = \max(0, x)$ 和 $f(x) = \ln(1 + e^x)$。将 ReLU 激活函数[10] 和 softplus 激活函数用于 RNN 中，相对于传统的 tanh 函数和 sigmoid 函数，可以处理梯度消失问题。因为在反向传播的过程中，误差信息的传播和激活函数的导数关系密切。误差值传播的简化表达式为：

$$E = \partial_E \cdot f'(net_1) \cdot W_1 \cdot f'(net_2) W_2 \cdots f'(net_n) \cdot W_n \quad (1)$$

如图 3 所示，sigmoid 函数的导数 f' 介于 0 到 0.25 之间，并且只在很窄的区间 $[-5, 5]$ 内数值较大，根据式(1)，误差值在反传学习过程中很容易衰减过快。tanh 函数的导数虽然介于 0 到 1 之间，但是数值下降得更快，也会面临一样的问题。而对于 ReLU 和 softplus 这一类型函数，其导数在相对宽广的区间内数值较大。其中 ReLU 函数的导数为 0 或 1，softplus 函数的导数则更为平滑。因此对于网络学习而言，相比 ReLU 函数 softplus 函数将为网络提供更大的可选择空间，提高模型的学习能力。

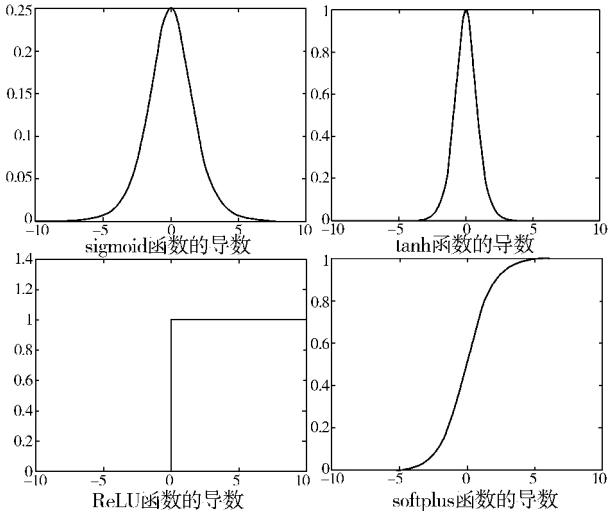

图 3　函数导数图

2　基于循环神经网络 3 种经典模型的改进

本文针对循环神经网络 3 种经典模型的改进主要涉及其中的激活函数部分。

对于传统 RNN 模型,ReLU 函数的使用有效提升了 RNN 的表现,缓解了梯度消失问题。同类型的 softplus 函数相较于 ReLU 函数更具有灵活性,可提高模型的学习能力。本文的改进方法是将 softplus 函数替换 RNN 的激活函数,提出新的模型 RNN + softplus。

对于含 LSTM 的 RNN 模型,其中的输出激活函数是其门机制中的重要结构,会直接影响模型的学习效果。传统的激活函数可能会限制该神经元的输出信息量,不利于信息的传递。本文提出基于 ReLU 函数和 softplus 函数来改进模型,以此提高模型的表达能力。LSTM 改进模型构建步骤如下:

Step1　构建遗忘门 $f_t = s(W_f \cdot [h_{t-1}, x_t] + b_f)$,初始化时其偏置量设为 1,即 $b_f = 1$,以减小在训练起始阶段遗忘信息过多[13]。

Step2　构建输入门 $i_t = s(W_i \cdot [h_{t-1}, x_t] + b_i)$,$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$,以及进行细胞更新 $c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$,该构建过程与传统模型一致。

Step3　在构建输出门时,将输出门 $o_t = s(W_o \cdot [h_{t-1}, x_t] + b_o)$,$h_t = o_t * \tanh(c_t)$ 中的 $h_t = o_t * \tanh(c_t)$,先后改进为 $h_t = o_t * \text{softplus}(C_t)$ 和 $h_t = o_t * \text{ReLU}(C_t)$。

GRU 中的候选激活函数是其门机制的输出部分。在设计上,GRU 同 LSTM 一样,使用了传统的 tanh 函数,因此限制了神经元的输出信息量。同 LSTM 类似,本文提出将 ReLU 函数和 softplus 函数用来改进 GRU 模型,探索不同类型的激活函数对门机制中输出部分的影响。GRU 改进模型构建步骤如下:

Step1　构建更新门 $z_t = s(W_z \cdot [h_{t-1}, x_t])$。

Step2　构建重置门 $r_t = s(W_r \cdot [h_{t-1}, x_t])$。以上构建方法均与传统模型相同。

Step3　在构建输出部分 $\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$,$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$ 时,将候选值 $\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$ 先后改进为 $\tilde{h}_t = \text{softplus}(W \cdot [r_t * h_{t-1}, x_t])$ 和 $\tilde{h}_t = \text{ReLU}(Wg [r_t * h_{t-1}, x_t])$。

本文所有模型的学习算法为目前广泛使用的截断 BPTT 算法,得到用于权重和偏置更新的误差偏导数。该算法包含 mini-batch 梯度下降方法,即在学习到每一组 mini-batch 样本序列后进行更新,相对于在线学习方法和全梯度学习方法能够更准确更快速地向收敛方向移动。

在初始化时,参数数值的合适与否直接关系到后期模型学习的好坏程度。因此,权重值采取了服从均匀分布的随机初始化。因为某 2 个节点权重值如果相同,则在有相同输入的情况下,根据学习算法的反向传播性质,这 2 个节点总会得到相同的梯度,也就难以学习到不同的特征。

为了防止模型过拟合,在模型配置技巧方面,epoch 轮数不能设置过大,该设置相当于早点结束策略。同时将权重值的数值范围限制在 0 附近比较小的区间内以创建一个更平滑的模型。此外进行梯度裁剪[13],以防止梯度值过大造成梯度爆炸,进而优化训练效果。

3　实验及结果分析

实验运行在 Windows 7 操作系统上的 VMware Workstation 10.0 虚拟机中,虚拟机的操作系统是 ubuntu 14.04 LTS,配有 5 GB 内存。本实验基于 TensorFlow 深度学习框架,在 Python 语言环境中进行搭建。TensorFlow 是 Google 在 2015 年 11 月发布的开源人工智能系统[14]。相对于其他深度学习框架例如 Caffe,torch 等,其通用性更好、功能齐全、社区异常活跃,有很好的前景。目前越来越多的研究人员从 torch 等框架转入 TensorFlow。但是目前 TensorFlow 只开源了单机版。

在实验中,需要配置的超参数如下: 初始化参数范围、学习率、梯度裁剪阈值、隐藏层数、隐藏层节点数、时间步数、迭代轮数、学习率开始衰减的迭代轮数、学习衰减率、小批量样本数。超参数设置的相关研究目前尚未取得突破,还没有明确而又广泛适用的结论。通常的做法是结合实际问题,通过实验用试算

的方法去寻找。本文就使用该种方法。

3.1 情感分析实验

本实验使用斯坦福大学的 Large Movie Review Dataset(LMRD) 情感分类数据集。LMRD 数据集包含 50 000 个样本,将其按 7:3 分为训练集和测试集。LMRD 数据集的影评词汇表有 20 000 个单词,该数据集获取方式可通过从斯坦福大学的网页上下载(http://ai. stanford. edu/ ~ amaas/data/sentiment/) 。模型评价的指标是测试集的准确率。

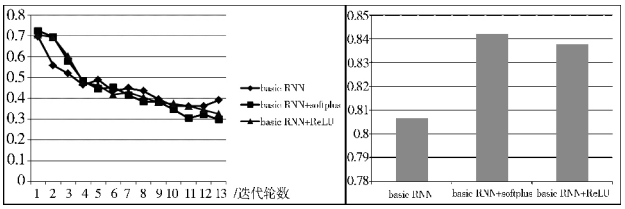

图 4 情感分析实验中传统 RNN 的 3 种模型实验结果对比图

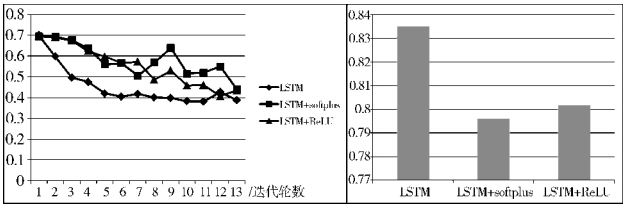

图 5 情感分析实验中 LSTM 的 3 种模型实验结果对比图

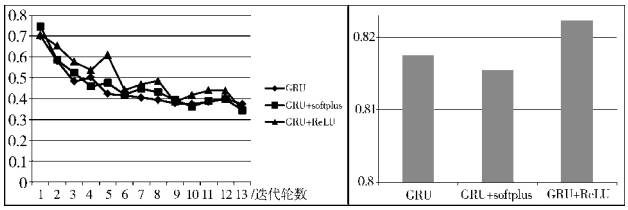

图 6 情感分析实验中 GRU 的 3 种模型实验结果对比图

通过实验对比分析发现,一共 9 个模型的准确率均在 79. 5% ~84. 2% 之间,可见循环神经网络比较适合于学习该数据集。如图 4 所示,改进后的 RNN + softplus 模型取得了最好的效果,收敛曲线平滑,测试集准确率为 84. 2%,优于 RNN + ReLU 和传统 RNN 模型。

在对门机制输出部分的改进中。GRU 和 LSTM 结构的各 3 个模型都在不同程度上出现收敛曲线波动。如图 5 所示,LSTM 结构的 3 个模型中传统 LSTM 效果最好,准确率达到 83. 5% 。LSTM + softplus 和 LSTM + ReLU 模型的测试集准确率相差值小于 0. 7% 。如图 6 所示,GRU 结构的 3 个模型相互间准确率差值小于 0. 6% 。其中 GRU + ReLU 模型的测试集准确率高于传统 GRU 模型和 GRU + softplus 模型。

3.2 语言模拟实验

本文使用 Penn Tree Bank(PTB) [16] 经典文本数据集进行文本层面的语言模拟。PTB 包含 929 000 个训练单词和 82 000 个测试单词,它的词汇表共有 10 000 个单词。PTB 数据集获取方式可从 Tomas Mikolov 的网站下载(http://www. fit. vutbr. cz/ ~ imikolov/) 。在配置超参数时,传统 RNN 的 2 个模型相对于另外 4 个模型,配置有较少的隐藏层节点和较小的学习衰减率。模型评价的指标是语言模型的困惑度。

困惑度是用来评价无标签聚类好坏程度的指标,常用于语言模型。困惑度的定义为:

$$\text{perplexity} = 2^{-l},\ l = \frac{1}{M}\sum_{i=1}^{m}\log\ p(\ s_i)$$

其中,$p(\ s_i)$ 为语言模型中语句的联合概率,M 为样本数目。由该定义可知困惑度数值越小,说明模型效果越好。

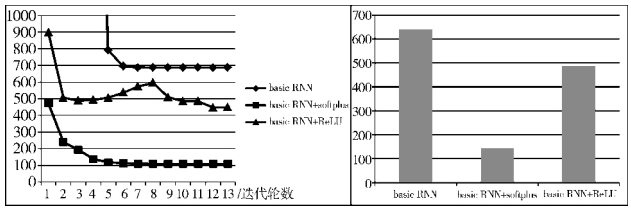

图 7 语言模拟实验中传统 RNN 的 3 种模型实验结果对比图

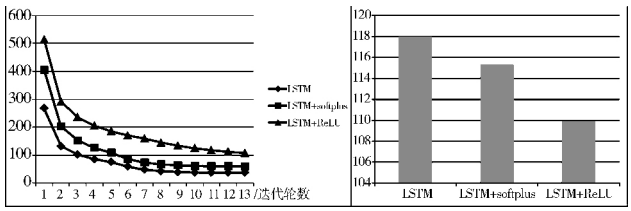

图 8 语言模拟实验中 LSTM 的 3 种模型实验结果对比图

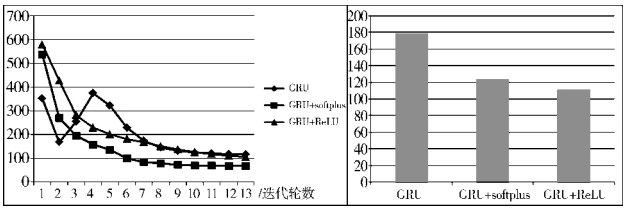

图 9 语言模拟实验中 GRU 的 3 种模型实验结果对比图

通过实验对比分析发现,改进后的 LSTM + ReLU 模型取得了最好的效果,收敛曲线平滑。如图 7 所示,对比传统 RNN 的 3 种模型,发现改进后的 RNN + softplus 模型取得了最好的效果,困惑度为 141. 875,优于 RNN + ReLU 模型。传统 RNN 模型内部比较简单,只有一个激活函数,因此激活函数的选取直接关系到模型学习的好坏。传统 RNN 模型内部由于是 tanh 函数,在学习过程中出现了梯度消失,造成学习效果很差。RNN + ReLU 模型虽然有了一些提高,但是也不是很理想。

在对门机制输出部分的改进中,改进后的模型效果均有提高。如图 8 所示,LSTM + softplus 和 LSTM + ReLU 模型的困惑度分别为 115. 284 和 109. 933,优于 LSTM 模型的困惑度 117. 899。如图 9 所示,

GRU + softplus 和 GRU + ReLU 模型的困惑度分别为 123.641 和 110.663，优于 GRU 模型的困惑度 178.732。因此 LSTM 类型的模型结构最适合在 PTB 数据集下进行文本生成学习。原有的门机制输出部分的功能相对较弱，ReLU 和 softplus 函数增强了包含门机制 RNN 模型的学习能力，特别是改进后的 GRU 模型效果提升比较明显。

此外，RNN + softplus 模型效果甚至优于 GRU 模型，可见改进后的传统 RNN 模型学习能力获得了大幅的提升。

4　结束语

本文探究了 2 种类型的激活函数对包含门机制 RNN 的作用，研究门结构中输出部分的激活函数对模型效果的影响。通过实验发现，改进后的模型效果有提升，在包含门机制的 RNN 输出部分中，ReLU 和 softplus 这一类型的函数可提高传统模型的表现。本文提出的 RNN + softplus 函数模型效果良好，甚至优于 GRU 模型。因此，在今后的 RNN 模型构建中可尝试将 ReLU 函数和 softplus 函数用于改进传统的激活函数，以此来处理梯度消失问题，同时在技巧配置方面控制梯度爆炸，可以有效学习到长时依赖知识。

受实验环境限制，参数调优还不够充分，也许存在更为理想的 RNN + softplus 模型参数配置。由于需要训练的网络结构较复杂，平均每个实验耗时近 40 小时，并且当网络规模增加时，极易造成内存溢出，因此下一步研究工作准备向分布式深度学习方向发展。

参考文献:

[1] 孙志军，薛磊，许阳明，等. 深度学习研究综述 [J]. 计算机应用研究，2012，29(8)：2806-2810.

[2] LeCun Y，Bengio Y，Hinton G. Deep learning [J]. Nature，2015，521(7553)：436-444.

[3] Rumelhart D E，Hinton G E，Williams R J. Learning representations by back-propagating errors [J]. Nature，1986，323(6088)：533-536.

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

[5] Gers F A，Schmidhuber J. Recurrent nets that time and count [C] // Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. 2000，3：189-194.

[6] Gers F A，Schmidhuber J，Cummins F. Learning to forget: Continual prediction with LSTM [J]. Neural Computation，2000，12(10)：2451-2471.

[7] Cho K，Merrienboer B V，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. 2014：1724-1734.

[8] Chung J，Gulcehre C，Cho K H，et al. Empirical evaluation of gated recurrent neural networks on sequence modeling [J]. arXiv preprint arXiv：1412.3555，2014.

[9] Graves A，Mohamed A，Hinton G. Speech recognition with deep recurrent neural networks [C] // 2013 IEEE International Conference on Acoustics，Speech and Signal Processing (ICASSP). 2013：6645-6649.

[10] Nair V，Hinton G E. Rectified linear units improve restricted boltzmann machines [C] // Proceedings of the 27th International Conference on Machine Learning. 2010：807-814.

[11] Le Q V，Jaitly N，Hinton G E. A simple way to initialize recurrent networks of rectified linear units [J]. arXiv preprint arXiv：1504.00941(2015).

[12] Greff K，Srivastava R K，Koutnik J，et al. LSTM：A search space odyssey [J]. arXiv preprint arXiv：1503.04069(2015).

[13] Mikolov T. Statistical language models based on neural networks [D]. PhD thesis：Presentation at Google，Mountain View，2012.

[14] Abadi M，Agarwal A，Barham P，et al. TensorFlow：Large-scale machine learning on heterogeneous distributed systems [J]. arXiv preprint arXiv：1603.04467，2016.

[15] Marcus M P，Marcinkiewicz M A，Santorini B. Building a large annotated corpus of English：The Penn treebank [J]. Computational Linguistics，1993，19(2)：313-330.

(上接第 28 页)

[12] He Jiang，Zhang Jingyuan，Xuan Jinfeng，et al. A hybrid ACO algorithm for the next release problem [C] // Proceedings of the 2nd International Conference on Software Engineering and Data Mining. 2010：166-171.

[13] Kumari A C，Srinivas K，Gupta M P. Software requirements optimization using multi-objective quantum-inspired hybrid differential evolution [J]. Advances in Intelligent Systems and Computing，2013，175：107-120.

[14] Sagrado J D，Águila D I M，Orellana F J. Multi-objective ant colony optimization for requirements selection [J]. Empirical Software Engineering，2015，20(3)：577-610.

[15] Souza J D T，Maia C L B，Ferreira T N D，et al. An ant colony optimization approach to the software release planning with dependent requirements [C] // Proceedings of the 3rd International Symposium on Search Based Software Engineering (SBSE'11). 2011：142-157.

[16] Jose J，Chaves-Gonzalez M，Miguel A，et al. Software requirement optimization using a multi-objective swarm intelligence evolutionary algorithm [J]. Knowledge-Based Systems，2015，83(7)：105-115.

[17] Askarzadeh A. Bird mating optimizer：An optimization algorithm inspired by bird mating strategies [J]. Communications in Nonlinear Science and Numerical Simulation，2014，19(4)：1213-1228.

[18] Zitzler E，Thiele L. Multiobjective evolutionary algorithms：a comparative case study and the strength Pareto approach [J]. IEEE Transactions on Evolutionary Computation，1999，3(4)：257-271.