

Character Level Full Convolution Neural Network for Text Classification

Sirui Li

School of Computer Science, Chengdu University of Information Technology, Chengdu Sichuan
Email: 635840708@qq.com

Received: Jan. 25th, 2020; accepted: Feb. 10th, 2020; published: Feb. 17^h, 2020

Abstract

In order to solve the problem of too many parameters in the full connection layer and low calculation efficiency of the traditional convolutional neural network, the full convolutional neural network and the global average pooling layer are used in image processing for text classification, the convolutional layer is combined with the global average pooled layer and the fully connected layer is replaced. Meanwhile, using the multi-scale convolution kernel with reference to the Inception structure reduces the number of parameters, speeds up the convergence, and increases the classification accuracy of the model. In addition, in order to avoid the curse of dimensionality and the slow speed of word level vector training, character level vector representation is used. And the batch standardization layer is used instead of the Dropout layer, reducing over-fitting problems. By using multiple indicators to evaluate the model in the test data set, the validity of the model is fully verified. Compared with the traditional model, the proposed model has better classification performance in the classification task.

Keywords

Text Classification, Character Level, Fully Convolutional Network, Global Average Pooling

字符级全卷积神经网络的文本分类方法

李思锐

成都信息工程大学, 计算机学院, 四川 成都
Email: 635840708@qq.com

收稿日期: 2020年1月25日; 录用日期: 2020年2月10日; 发布日期: 2020年2月17日

摘要

为了解决传统卷积神经网络的全连接层参数过多, 计算效率低的问题。该文将图像处理中使用的全卷积神经网络和全局平均池化层用于文本分类, 将卷积层和全局平均池化层结合并替换全连接层, 同时参照 Inception 结构使用多尺度的卷积核, 减少了模型的参数数量, 加快了模型的收敛速度, 增加了模型分类准确率。此外为了避免维度灾难和词级向量训练速度慢的问题, 该文采用字符级进行向量化表示。并使用批量标准化层代替 Dropout 层, 减少了过拟合问题。通过使用多指标在测试数据集中进行模型评估, 充分验证了该模型的有效性, 与传统模型相比, 提出的模型在分类任务中具有更好的分类性能。

关键词

文本分类, 字符级, 全卷积神经网络, 全局平均池化层

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着信息技术的不断发展, 目前互联网中每天流动的信息数量与日俱增, 其中文本数据占据了很大一部分。过去人们通过纸质媒介来记录文本信息, 文本信息数量少, 仅通过人工的方式就能很好地完成文本分类任务。但如今大数据时代, 文本的存储介质变成了硬盘、数据库等, 光是每天新产生的文本数量就早已超过了人工处理的极限。所以需要借助文本分类技术从海量信息中提取有效的信息, 来帮助人们学习和工作。

文本分类一直都是自然语言处理(NLP)中的经典问题, 传统分类技术最早可以追溯到上世纪 50 年代的专家规则(Pattern)分类方法。80 年代初专家规则发展到利用知识工程建立的专家系统, 但是专家系统分类效率较低, 适用的范围和准确率都非常有限。后来伴随着统计学习方法的发展, 机器学习的兴起, 出现了很多稳定、分类效率高的传统方法, 例如 Rocchio 分类, 逻辑回归(LR)、NaïveBayes 分类器(NBC)、k-最近邻(KNN)、支持向量机(SVM)和基于树的决策树和随机森林等。但大数据时代文本长度较短, 特征稀疏, 有很多奇特的符号和拼写错误的词语, 传统方法已经无法满足该时代的文本分类要求。

近年来, 深度学习发展迅速, 特有的层次结构能很好地从浅层特征中提取高层特征, 已经在图形图像、自然语言处理等领域取得了重大突破, 成为了机器学习的一个新兴领域。其中卷积神经网络因为其能够进行卷积运算, 高效地利用数据, 同时对序列处理特别有效的特点, 在文本分类方面得到了很高的关注度, 各种优化模型层出不穷。但是文本分类中的卷积模型全都受限于全连接层, 无法避免全连接层参数过多, 容易引起过重复计算和参数冗余的问题。所以本文提出一种替换全连接层的全卷积模型, 通过对比传统卷积神经网络和其他的文本分类模型表明, 我们的方法拥有更好的分类性能。

论文组织结构如下: 第二节介绍相关工作的背景知识。第三节详细介绍提出的全卷积神经网络分类模型。第四节给出实验结果及分析。最后一节对本文工作进行总结并介绍未来研究的方向。

2. 相关工作

2006 年 Hinton [1] 等人利用逐层贪心算法初始化深度信念网络(DBN), 提出了深层学习的概念。此后

很多的深度学习模型被提出, He [2]等人和 Abdel-Hamid [3]使用深度神经网络分别在计算机视觉领域和语音识别方面取得了显著的成果。

近些年,深度学习概念开始广泛应用于自然语言处理领域,如 Kim [4] [5]、Kalchbrenner [6]等人分别改进了卷积结构,提出了不同结构的文本分类模型,在多个数据集上取得了比较理想的结果。Zhang [7]等人将字符作为一种文本的新特征输入卷积神经网络,并应用于文本分类,证实了字符级的有效性。Poon [8], 冯兴杰[9], 何炎祥[10]等人通过结合卷积神经网络和循环神经网络,在文本的情感分析任务上取得也了出色的结果。除此之外, Zhou [11]提出的 cnn 与 lstm 结合的 c-lstm 模型和 Joulin [12]等人提出的 FastText 模型,在文本分类任务中都有很好的性能。

在全卷积神经网络(FCN)方面,2014年 Min Lin [13]等人在论文中改进了 Cnn 网络模型,首次提出了抽象能力更高的 Mlpconv 层和全局平均池化(Global Average Pooling)的概念,提高了卷积模型的特征提取能力,减少了网络模型的参数,这篇论文为全卷积网络和 Inception 结构的提出和发展提供了一个很好的开端。2014年 Szegedy [14]等人提出了 GoogLeNet 模型,该模型首次使用 Inception 结构代替了单层的卷积加激活的传统操作,使模型精度上升的同时加快了训练速度。2015年 Jonathan Long [15]的论文首次提出全卷积概念,并将其应用于图像语义分割,解决了图像语义分割无法端到端的问题,避免了像素的重复存储和计算。到2017年底,FCN已经分化出了数百种的模型结构。FCN使用卷积层替换掉了传统 CNN 的全连接层,因为卷积层能学习特征之间的联系,并且卷积层参数较少,收敛迅速、稳定且计算速度快。正因为 FCN 有上述的优点,所以部分人将 FCN 用作其他深度学习任务,如张曼[16]等人将全卷积神经网络用作文本分类任务,证明了 FCN 在文本分类任务上有加快模型收敛速度,减少模型复杂度等优势。

中文等语言采用词级别的深度神经网络时,词语通常是采用词嵌入的思想将词语转化为低维稠密的向量来避免维度灾难,再将文本表示成为一个二维的矩阵输入给模型训练。训练词向量的工具有 Google 公司中 Tomas Mikolov [17]等人推出的 Word2vec 工具, Pennington [18]等人提出的 GloVe 模型以及 Matthew E. Peters [19]提出的基于动态词向量训练的 ELMo 等。因为所获得的词向量包含一定的句法结构和语义结构的信息,模型所获得的分类效果颇佳,缺点是词向量依赖优秀的词汇库,如果词汇库不够优秀和完善,就会遇到 OOV (Out of Vocabulary)问题,影响模型的最终输出。而采用字符级别的神经网络,优势是不需要使用预训练好词汇库信息,极大地提升模型所能处理的词汇量,避免了词汇库不完善的问题。并且能弹性地处理拼写错误和罕见词,同时降低模型的计算成本,加快模型的训练速度。除此之外,字符级可以很容易的应用于所有语言和数据集,具有很高的泛化性。

综上所述,许多学者已经提出了很多优化模型,取得了优异的成绩。但到目前为止,模型的优化不再只追求准确率的提升,而是速度和准确率的平衡。所以在保持准确率上升的同时提升速度是关键的问题。

3. 模型设计与算法流程

为了有效地提高文本的分类效率,本文采用全卷积神经网络,使用卷积层和全局平均池化层代替全连接层,结合全局平均池化层减少参数,卷积层收敛迅速、稳定且计算速度快的优势。因为去除了全连接层,所以传统卷积神经网络的 Dropout 层无法发挥太大的作用,本文在卷积层后加入批量标准化层(Batch Normalization)来代替 Dropout 层防止过拟合,同时使用 Inception 结构增加分类的准确率。

3.1. 总模型

本文设计的字符级全卷积神经网络(Inception-CharFcn)如图 1 所示,将卷积神经网络的全连接层替换为全局平均池化层加全卷积层,同时采用多尺度卷积核的思想,使用两层 Inception 的结构,最深处有 4

层卷积。设置了两个 loss 单元，在中间层加入了辅助计算的 loss 单元，目的是计算损失时让第一层的特征也有很好的区分能力，帮助网络的收敛，从而让网络能更好地训练。第一层 Embedding 层将输入的中文数据进行字符级嵌入操作，转换为向量矩阵，然后进入第一个 Inception 结构，分别由多个不同尺寸的卷积核进行卷积运算，合并特征向量后进入下一个 Inception 结构，再次合并后将激活的张量输入全局平均池化层(Global average pooling)，最终由全卷积层(Fcn)继续进行两次卷积运算并输出分类结果。输出层的长度是由具体的类别数目决定的。

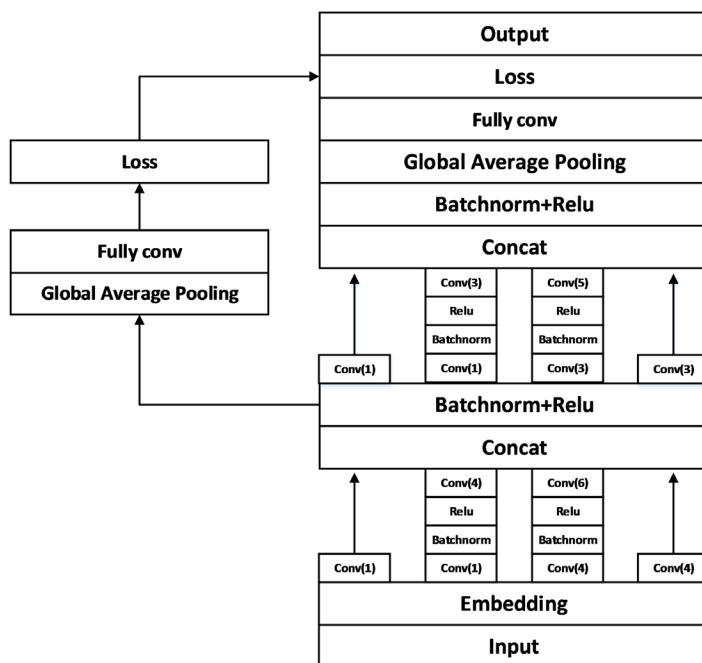


Figure 1. Overall structure of the model

图 1. 模型总体结构

3.2. 全卷积和全局平均池化

全卷积神经网络是由卷积层代替传统卷积神经网络中的全连接层来实现的，因为全连接层可以看做一个特殊的卷积层，所以该神经网络将全连接层替换为等价的卷积层。其中传统卷积层的时间复杂度为公式 1 所示， M 是输出特征向量(Feature Map)的尺寸， K 为卷积核(Kernel)的尺寸， C_{in} 为输入通道数， C_{out} 为输出通道数。而在公式 2 的单个卷积层的空间复杂度公式中可以看到，卷积层的空间复杂度只与卷积核的尺寸 K 、通道数 C 相关，而与输入图片尺寸无关。

$$\text{Time} \sim O(M^2 * K^2 * C_{in} * C_{out}) \quad (1)$$

$$\text{Space} \sim O(K^2 * C_{in} * C_{out}) \quad (2)$$

但是将全连接层看做一个特殊的卷积层时，其卷积核尺寸 K 与输入矩阵 X 一样，每个卷积核输出特征图是一个标量点，即 $M = 1$ ，如公式 3 和公式 4 所示，全连接层的空间复杂度与输入数据的尺寸密切相关。因此如果输入数据尺寸越大，模型的体积也就会越大。

$$\text{Time} \sim O(1^2 * X^2 * C_{in} * C_{out}) \quad (3)$$

$$\text{Space} \sim O(X^2 * \text{Cin} * \text{Cout}) \quad (4)$$

由上述公式可以看出，如果直接将全连接层替换为卷积层，模型的大小受到输入数据尺寸的影响，参数并不会减少。而使用全局平均池化层来替换全连接层，会使模型直接降维，极大地减少模型的参数量，但是为了实现该过程，需要将最后一层卷积的输出通道数对应具体的分类类别数目，虽然这相当于直接赋予了每个输出通道实际的类别意义，但是这会导致模型的收敛速度变慢。

虽然全连接层可以看做一个特殊的卷积层，但是之间还是存在着区别。当模型训练完成时每个连接都拥有权重，其中全连接层会给每个连接分配权重但不会改变连接关系，而卷积层则会学习目标和那些特征之间的关系，有用的关系保留，没用得到关系它会弱化或者直接抛弃，同时全部卷积层可以共用一套权重，减少了重复计算，降低了模型复杂度，使模型收敛更加迅速。

所以本文使用结合全卷积和全局平均池化的结构，在不改变最后一层卷积的输出通道数的情况下，先利用全局平均池化将数据的尺寸变为 1，再将数据输入全卷积层进行分类。这样不仅利用全卷积结构收敛速度快快的特点解决了单一全局平均池化收敛速度缓慢的问题，同时由公式 5 和公式 6 可知，特征向量大小变为 1 后全卷积层复杂度只与输入、输出通道数相关，模型参数量极大减少，运算速度加快。

$$\text{Time} \sim O(\text{Cin} * \text{Cout}) \quad (5)$$

$$\text{Space} \sim O(\text{Cin} * \text{Cout}) \quad (6)$$

3.3. 批量标准化

因为本文模型中没有全连接层，而 Dropout 在卷积层后的作用又不明显，所以为了归一化输入增加模型的泛化能力，我们在卷积层后添加了批量标准化层(Batch Normalization)。批量标准化(Batch Normalization)是 Sergey Ioffe 和 Christian Szegedy [20]在 2015 年的论文中提出，解决小批量随机梯度下降算法在训练过程中输入分布多变的问题。批量标准化首先对每个 batch 的输入 $C = \{X_1, X_2, \dots, X_m\}$ 求平均值，其中 batch size 等于 m ，然后计算该 batch 每个输入的均值 μ 和方差 σ 。再然后对于所有的 $X_i \in C$ 进行一个标准化，把取值区间逐渐向极限饱和和区靠拢的输入分布拉回到均值为 0、方差为 1 的比较标准的正态分布，使得输入值落入比较敏感的区域，以此避免梯度消失问题，同时增大导数值，增强反向传播的流动性，加快训练收敛速度。但是上述标准化过程会导致网络的表达能力下降，为了防止这一点，最后每个神经元增加两个调节参数(γ 和 β)，这两个参数是通过训练来学习到的，通过反向变换标准化后的输入值，使得网络表达能力增强。具体计算如下列公式 7 至 10 所示：

$$\mu \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad (7)$$

$$\sigma_c^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_c)^2 \quad (8)$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_c}{\sqrt{\sigma_c^2 + \varepsilon}} \quad (9)$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta(x_i)} \quad (10)$$

3.4. 字符表示

本文使用中文的单字作为字符向量，相较于单词级的嵌入，字符级别的嵌入能极大缩小字典的长度，降低计算成本和提升计算速度，同时能减少处理拼写错误和罕见词对分类的影响。模型的第一层为字符

向量嵌入层，目的是将输入数据的向量表示同主分类任务一起进行训练，共享神经网络的权重。进行向量嵌入的第一步是对文本进行分词操作，同时去除停用词。第二步将分好词的文本按单个字的出现次数排列，并建立对应字典。

第三步建立向量嵌入层，创建一个行列分别是最大单字索引数和嵌入维度的向量矩阵，第四步对应建立的字典，将整数索引映射为字符向量，输出一个字符向量向量矩阵，最后将该矩阵输入模型进行训练。

3.5. 算法实现

算法实现步骤如下。第一步，将输入的文本按照上文描述的字符级嵌入方法进行向量化表示，使整个文本形成一个向量矩阵 $A \in R^{s \times d}$ ，其中 s 为固定后的文本长度， d 为字符嵌入向量的维度。第二步，将 A 输入模型与第一层 Inception 结构中的多尺度卷积核进行卷积运算，得到基于不同卷积核的特征向量 V_i ， i 代表不同的卷积核。然后应用 1-Max 池化来提取特征向量 V_i 中的最大值，所有最大值合成一个卷积最终特征 $B_i \in R^k$ ，其中 k 为卷积核的数量， i 代表不同卷积核。第三步，将不同卷积核的最终卷积特征 B_i 合并为一个新特征 $X \in \text{concat}(B_i)$ 后将新特征输入批量标准化层进行归一化和 ReLU 非线性激活，输出特征为 D_1 。第四步，将 D_1 输入第二层 Inception 结构，重复第二步和第三步，输出特征为 D_2 。第五步，将 D_2 输入全局平均池化层，使特征向量大小变为 1，形成一个新的向量矩阵 $E \in R^{1 \times b}$ ， b 为构成新特征的卷积核数量之和。然后将新的矩阵 E 放入全卷积层，继续进行两次卷积运算，最后输出特征 $F \in R^y$ ，该特征就是模型的预测值，其中 y 为最终分类的类别数。第六步，将特征 F 通过 argmax 函数得到得分最高的项，该项就是分类预测，最后通过交叉熵计算损失函数(Loss function)用来训练模型，但不同于传统模型，本文设置的辅助 loss 单元会输出由 D_1 得到的辅助损失函数，该损失函数乘以 0.5 后和后面的损失函数值相加作为最终的损失函数来训练网络。其中损失函数的运算如公式 11 所示， p 为分类的标签， f 为预测值。

$$\text{loss}(p, f) = -\frac{1}{n} \sum_{i=1}^n p(x_i) \log(f(x_i)) \quad (11)$$

4. 实验

4.1. 超参数设置

在本文模型中，输入的是经过嵌入处理的字符向量。神经网络每批训练大小为 64，总共迭代 20 个 epoch，每训练 100 个时间步验证一次性能，并保存最好结果。激活函数设置为 ReLU，学习率设置为 $1e-3$ ，若 1000 轮迭代性能未提升则衰减为一半，部分参数如表 1 所示。本文模型设置了 5 种尺度的卷积核，分布在 2 层 Inception 结构中。2 层代替全连接层的卷积核大小为 1，部分卷积具体结构如表 2 所示。

Table 1. Experimental parameters and specifications

表 1. 实验超参数与规格

参数名	含义	参数取值
batch_size	每批训练大小	64
num_epochs	最大训练轮数	20
learning_rate	学习率	$1e-3$
embedding_dim	词向量维度	128
vocab_size	词汇表大小	5000

Table 2. Convolution kernel structure**表 2.** 卷积核结构

卷积核名称	卷积尺寸	卷积核数量
kernel_size1	1	256
kernel_size3	3	256
kernel_size4	4	256
kernel_size5	5	256
kernel_size6	6	256

4.2. 评价标准

对模型的性能评估需要有效的评估标准，本文分类效果的评估方法分为两种，第一种是对分类任务中不同分类标签的精度(Precision)、召回率(Recall)和 F1 值进行宏观平均和加权平均计算，用来衡量模型在准确率方面的性能。第二种是记录每 10 个时间步模型的 loss 值和准确率(Accuracy)，用来衡量模型在收敛速度方面的性能。

其中精度是分为一类的样本中，正确分类的样本的比例；召回率是该类正确分类的样本与该类全部样本的比例，衡量的是分类系统的分类完整率。F 值是召回率和准确率的加权调和平均，因为召回率和准确率在文本分类中同等重要，所以常用的为 F1 值。损失函数(Loss function)是用来估量模型的预测分类与真实分类的不一致程度。准确率(accuracy)是所有样本中分类正确样本所占的比例。

4.3. 实验数据

本文搜集了 4 种中文数据集，包含长文本和短文本，所有数据都经过分词处理。

复旦大学数据集：该数据集由复旦大学自然语言处理小组收集整理，用于训练文本分类模型。其中 test_corpus 为测试语料，共 9833 篇文档；train_corpus 为训练语料，共 9804 篇文档，两个预料各分为 20 个相同类别，20 个类别文件夹下存放着对应类别的文档。

THUCnews 新闻数据集：该数据集由清华大学自然语言处理实验室搜集与整理，通过过滤筛选 2005~2011 年间新浪新闻数据生成，包含 74 万篇新闻文档，均为 UTF-8 纯文本格式，并重新整合划分出彩票、教育、娱乐、科技等 14 个分类类别，所有文档均按对应类别储存在 14 个类别文件夹下。

搜狐新闻数据集：该数据集来自搜狐新闻 2012 年 6 月~7 月期间网站 18 个频道的新闻数据，所有分类混合保存在 128 个文档中，由搜狐实验室收集整理。数据格式为 url 网页代码，需要进行额外的处理。

头条新闻数据集：该数据集拥有 15 个分类共 38w 条数据，数据来源于头条新闻，单条数据长度较短，适合用来训练短文本的模型分类。

为了增加数据集的有效性和丰富性，我们将上述 4 个数据集中相同或相似分类的数据随机抽取组合构成一个新数据集，新数据集共有财经、房产、教育、科技、运动、游戏、娱乐、军事、旅游、文化 10 个分类，每个分类 6000 条数据，按照 7:2:1 的比例分为训练集、测试集、验证集，并将文本每一条数据统一为分类标签 + /t + 文本的方式。在进入模型训练前进行乱序操作，保证实验的真实性和泛化能力。具体如表 3 所示。

Table 3. Data sample distribution**表 3.** 数据样本分布

	训练集	测试集	验证集
样本数	42000	12000	6000

4.4. 实验与结果分析

4.4.1. 对 Inception 结构和全卷积效果的验证

为了体现 Bn 层、Inception 结构和全卷积对模型的性能提升有不同的作用,我们分别设置了 charcnn, 添加了 bn 层的 bn-charcnn, 将 charcnn 全连接层替换为全卷积和全局平均池化的 CharFcn, 添加了 Inception 结构的 Inception-charcnn, 以及本文提出的添加了 Inception 结构的 Inception-CharFcn, 共 5 个模型在上文构造的新数据集上进行对比实验, 5 个模型除了 charcnn 都添加了批量标准化层。表 4 中列出的是不同模型在准确率、召回率和 f1 值上的对比结果, 其中准确率保留小数点后两位, 召回率和 f1 值为每个分类的加权平均只保留整数位。图 2 和图 3 列出的是 Charcnn、CharFcn、Inception-charcnn、Inception-CharFcn, 4 个模型在新数据集的训练过程中准确率和 Loss 值随时间变化的曲线图。

总结上述数据可知, 添加 bn 层、Inception 结构和全卷积的模型在数据集上都获得了性能的提升。添加了批量标准化层的 bn-charcnn 相较于 charcnn 准确率明显上升。而使用了全卷积层的 charfcn 虽然准确率逊色于 Bn-charcnn, 但是其收敛速度和准确率相较于 Charcnn 都有明显的提升。使用了 Inception 结构的 Inception-CharFcn 和 Inception-CharCnn, 对比没有使用该结构的模型准确率有明显提升, 且收敛速度加快。而 Inception-CharFcn 与 Inception-CharCnn 对比, 虽然准确率的提升并不明显, 但是在训练时 Inception-CharFcn 模型的准确率和损失函数的收敛速度都有加快, 对缩短训练时间有帮助。

Table 4. Comparison of test results of the model

表 4. 模型的测试结果对比

模型	准确率(%)	召回率(%)	F1 值(%)
charcnn	85.43	85	85
Bn-charcnn	87.60	88	88
CharFcn	87.44	87	87
Inception-CharCnn	88.83	89	89
Inception-CharFcn	88.89	89	89

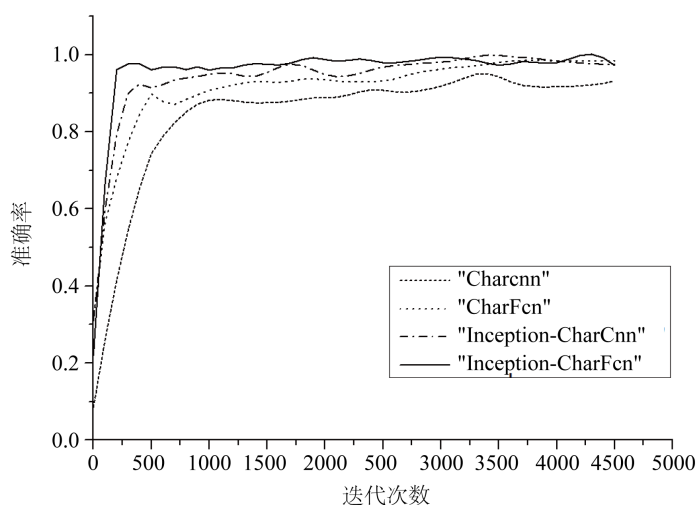


Figure 2. Curve of accuracy rate with time during training

图 2. 训练过程中准确率随时间变化曲线

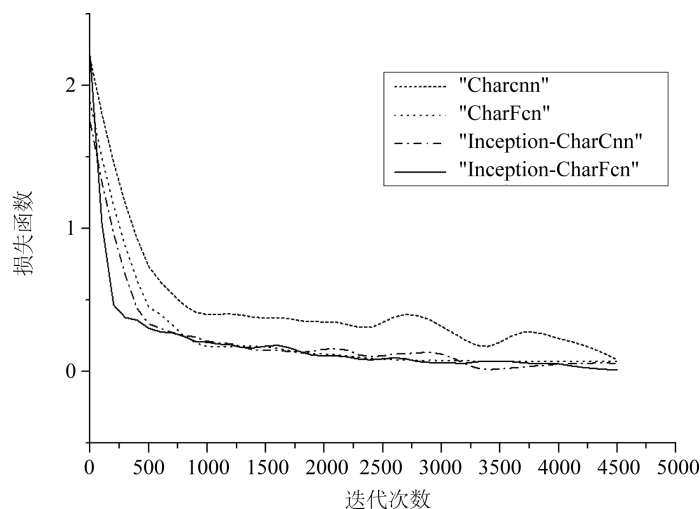


Figure 3. Curve of loss with time during training
图 3. 训练过程中损失随时间变化曲线

4.4.2. 对比实验

为了从更多的角度来验证本文提出的神经网络模型在中文文本分类中的有效性和适用性，本文选取了多个模型进行在 THUCnews 的中文数据上进行对比实验，其中传统文本分类方法选择 logistic 回归和随机森林(Random Forest)，神经网络模型选用 c-lstm、textcnn、textrnn、twoCNNTxtRelation、FastTest、BERT，6 种在多种 nlp 任务中表现优异的神经网络模型与改进模型。由表 5 中的实验数据对比可知，在与上述模型的比较中，本文模型在准确率上拥有不逊于其他优秀分类模型的准确率，并超过了传统分类方法和一些神经网络。

可以得出结论，本文提出的 bn-charfcn 在文本分类任务中表现良好，同时能加快 cnn 的收敛速度，且表现优于传统方法，值得深入研究。

Table 5. Comparison of accuracy between Inception-CharFcn model and other models

表 5. Inception-CharFcn 模型与其他模型准确率对比

模型	准确率
logistic 回归	94.23
随机森林	92.28
c-lstm	95.78
textcnn	96.33
textrnn	93.83
twoCNNTxtRelation	96.07
FastTest	95.51
BERT	96.96
Inception-CharFcn	96.86

5. 结束语

在本文中, 我提出了一种适用于中文文本分类的神经分类方法。该模型应用了字符级卷积神经网络(CharCnn)、全卷积神经网络(Fcn)、全局平均池化(Global average pooling), 以及 Inception 结构, 分别利用各自的优势, 达到了预期的效果。从实验结果的分析来看, 该模型拥有较快的收敛速度和令人满意的分类性能。同时由于使用字符级卷积网络, 该模型在面对不同的数据集和语言时, 也有较强的适用性。所以可以得出结论, 本文提出的模型优于标准 cnn 模型和传统分类模型, 与其他改进方法也有可比性。

对于未来的研究方向, 本文模型虽然使用了 Inception 结构, 但最深处只有 4 层卷积, 未来可以朝着更宽更深的方向前进, 同时可以使用更加优秀的改进卷积核, 来进一步减少模型的参数数量, 增加模型的分类效率。模型的超参数也可以进一步地进行优化, 使其更加稳定, 分类性能也会更好。

基金项目

四川省科技厅重点项目(2017GZ0331)。

参考文献

- [1] Hinton, G.E. and Salakhutdinov, R.R. (2006) Reducing the Dimensionality of Data with Neural Networks. *Science*, **313**, 504-507. <https://doi.org/10.1126/science.1127647>
- [2] He, K., Zhang, X., Ren, S., et al. (2016) Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 770-778. <https://doi.org/10.1109/CVPR.2016.90>
- [3] Abdelhamid, O., Mohamed, A., Jiang, H., et al. (2012) Applying Convolutional Neural Networks Concepts to Hybrid NN-HMM Model for Speech Recognition. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, 25-30 March 2012, 4277-4280. <https://doi.org/10.1109/ICASSP.2012.6288864>
- [4] Kim, Y. (2014) Convolutional Neural Networks for Sentence Classification. *Proceeding of the Conference on Empirical Methods in Natural Language Processing*, Doha, 16. <https://doi.org/10.3115/v1/D14-1181>
- [5] Kim, Y., Jernite, Y., Sontag, D., et al. (2015) Character-Aware Neural Language Models. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, 12-17 February 2015, 2741-2749.
- [6] Kalchbrenner, N., Grefenstette, E. and Blunsom, P. (2014) A Convolutional Neural Network for Modelling Sentences. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, 23-25 June 2014, 655-665. <https://doi.org/10.3115/v1/P14-1062>
- [7] Zhang, X., Zhao, J. and Lecun, Y. (2015) Character-Level Convolutional Networks for Text Classification.
- [8] Poon, H.K., Yap, W.S., Tee, Y.K., et al. (2018) Document Level Polarity Classification with Attention Gated Recurrent Unit. *International Conference on Information Networking*, Chiang Mai, 10-12 January 2018, 7-12. <https://doi.org/10.1109/ICOIN.2018.8343074>
- [9] 冯兴杰, 张志伟, 史金钊. 基于卷积神经网络和注意力模型的文本情感分析[J]. 计算机应用研究, 2018(5): 1434-1436.
- [10] 何炎祥, 孙松涛, 牛菲菲, 等. 用于微博情感分析的一种情感语义增强的深度学习模型[J]. 计算机学报, 2017, 40(4): 773-790.
- [11] Zhou, C., Sun, C., Liu, Z., et al. (2015) C-LSTM Neural Network for Text Classification. *Computer Science*, **1**, 39-44.
- [12] Joulin, A., Grave, E., Bojanowski, P., et al. (2016) Bag of Tricks for Efficient Text Classification. <https://doi.org/10.18653/v1/E17-2068>
- [13] Lin, M., Chen, Q. and Yan, S. (2013) Network in Network.
- [14] Szegedy, C., Liu, W., Jia, Y., et al. (2014) Going Deeper with Convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 1-9. <https://doi.org/10.1109/CVPR.2015.7298594>
- [15] Long, J., Shelhamer, E. and Darrell, T. (2014) Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **39**, 640-651.
- [16] 张曼, 夏战国, 刘兵, 周勇. 全卷积神经网络的字符级文本分类方法[J/OL]. 计算机工程与应用, 1-11. <http://kns.cnki.net/kcms/detail/11.2127.TP.20190327.1747.010.html>, 2019-10-05.
- [17] Mikolov, T., Chen, K., Corrado, G., et al. (2013) Efficient Estimation of Word Representations in Vector Space.

-
- [18] Pennington, J., Socher, R. and Manning, C. (2014) Glove: Global Vectors for Word Representation. *Conference on Empirical Methods in Natural Language Processing*, Doha, 1532-1543.
 - [19] Peters, M.E., Neumann, M., Iyyer, M., *et al.* (2018) Deep Contextualized Word Representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1, 2227-2237.
 - [20] Ioffe, S. and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *International Conference on Machine Learning*.