基于改进自编码器的文本分类算法

许卓斌 郑海山 潘竹虹
(厦门大学信息与网络中心 福建 厦门 361005)

摘 要 词的向量化表达是文本挖掘应用的必要前提。为了改善自编码器在词嵌入中的效果,提高文本分类的准确性,提出了一种改进的自编码器并将其用于文本分类。在传统自编码器的基础上,在隐藏层加入了一个全局调整函数,其将绝对值小的特征值调整到绝对值大的特征值上,实现了隐藏层特征向量的稀疏化。得到调整后的特征向量之后,采用全连接神经网络进行文本分类。在20news数据集上的实验结果表明,所提方法具有更好的词向量嵌入式效果,并且在文本分类中也具有更好的效果。

关键词 文本挖掘,自编码器,嵌入式向量,神经网络

中图法分类号 TP391.4 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.06.037

Improved Autoencoder Based Classification Algorithm for Text

XU Zhuo-bin ZHENG Hai-shan PAN Zhu-hong
(Information and Network Center, Xiamen University, Xiamen, Fujian 361005, China)

Abstract Vector representation of words is the premise of applications in text mining. In order to improve the effectiveness of autoencoders in words embedding and the accuracy of text lassification, this paper proposed an improved autoencoder and applied it for text classification. Based on traditional autoencoder, a global adjustable function is added to the latent layer, which adjusts smaller absolute values to bigger absolute values and implements the sparsity of characteristic vector in the latent layer. With the adjusted latent characteristic vector, a full connected neural network is used to classify text. The experiments on 20news dataset show that the proposed method is more effective in words embedding, and has better performance in text classification.

Keywords Text mining, Autoencoder, Embedding vector, Neutral network

1 引言

在文本数据挖掘中,文本分类是一项重要的研究内容,被广泛应用于Web搜索、日志分析、信息过滤、情感分析等领域中。对文本数据进行分类的前提是提取文本信息的特征并将其向量化,常用的方法有词袋模型(Bag of Words,BoW),如一元、二元和n元模型等;此外,还可以采用基于频率的特征提取方法,如LSA[1],PLSA[2],以及LDA[3]等。然而,上述特征表达方法均忽略了单词在句子中的上下文信息,也没有考虑单词在句子中的顺序,因而不能准确地捕获单词的语义信息。在n元模型中[4],当n取值较大时可以捕获单词的语义信息,然而它们都面临数据稀疏的问题,其分类应用的准确性也不高。

近年来,词嵌入与深度神经网络的发展对自然语言处理的研究产生了巨大的影响。词嵌入是语料库中单词的分布式表达,这种方法可以很好地缓解文本挖掘中的数据稀疏问题。Mikolov等人[5]提出了著名的词嵌入方法Word2Vec,该方法基于句子级别对单词进行向量化,能捕获单词在句子中的结构和语义等信息,还可以解决文本挖掘面临的数据稀疏性问题,在许多文本挖掘应用中都表现出了很好的效果。此外,在Word2Vec的基础上,Le等人[6]又提出了针对整篇文档的词嵌入方法Doc2Vec。

自编码器[7]是一种神经网络结构,它将输入数据向量化,然后通过最小化重建输入数据的误差对特征进行提取。在自编码器中,如果输入数据是文本,输出数据是对输入的重建,那么就可以将中间的嵌入式向量理解为输入数据的特征向量。自编码器一经提出,便引起了广泛的关注。基于该自编码器,研究人员提出了许多变种,如去噪音自编码器[8]、变分自编码器[9]、稀疏自编码器[10]等。自编码器在图像处理领域取得了很好的效果,如MNIST[11]和CIFAR[12]。然而,由于文本数据具有纬度高和稀疏性等特征,因此自编码器在文本挖掘领域的应用效果还有待提高。

为了解决应用自编码器进行文本嵌入的过程中面临的纬度高和数据稀疏性问题,以提高其在文本分类应用中的效果,

到稿日期:2018-02-28 返修日期:2018-04-20 本文受赛尔网络下一代互联网技术创新项目(NGII20160410)资助。

许卓斌(1975-),男,硕士,高级工程师,主要研究方向为数据中心、园区网、高校信息化应用建设、虚拟化、并行计算、大数据,E-mail:zbxu@xmu.edu.cn(通信作者);**郑海山**(1979-),男,硕士,高级工程师,主要研究方向为大数据、校园信息化建设与管理;**潘竹虹**(1982-),女,硕士,高级工程师,主要研究方向为园区网络管理、网络日志、智能分析。

提出了一种改进的自编码器来进行文本向量的嵌入，并应用嵌入后的向量进行文本分类。

2　改进的自动编码器

自动编码器包含输入层、隐藏层和输出层，输入层和隐藏层之间是全连接神经网络，隐藏层和输出层之间也是一个全连接的神经网络。输入层的输入数据是向量化后的文本数据，如 one-hot 向量[13]；隐藏层可被视为神经网络对输入向量提取出的特征向量；输出层得到的输出数据是神经网络对特征向量进行变化得到的输入数据的重建；当输入数据和输出数据一致时（即成功重建输入数据），隐藏层的向量就可看作是输入数据的特征向量。

本文对传统的自编码器进行了改进，在隐藏层中加入了一个全局调整函数 g。通过函数 g，将隐藏层的特征向量进行稀疏化，使其仅保存若干个非 0 值。该网络的结构如图 1 所示。

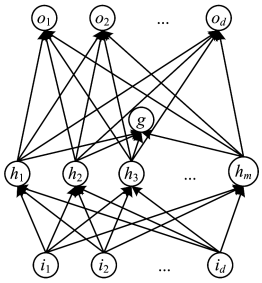

图 1　改进的自动编码器的结构图

Fig. 1　Structure of improved autoencoder

在图 1 给出的改进的自编码器结构中，向量 i 为 d 维的输入向量，向量 h 为隐藏层的 m 维特征向量，向量 o 为 d 维的输出向量，g 为 h 的全局调整函数（工作原理将在下文描述）。

给定输入向量 i，通过全连接神经网络计算得到的隐藏层向量为：

$$h=\tanh(Wx+b) \tag{1}$$

其中，W 为下层神经网络的参数矩阵，b 为隐藏层向量 h 的偏差常量，$\tanh(\cdot)$ 为双曲正切函数：

$$\tanh(x)=\frac{e^{2x}-1}{e^{2x}+1} \tag{2}$$

得到隐藏层向量 h 后，采用全局调整函数 g 对 h 进行稀疏化。其稀疏化过程为：

1）按照正负将 h 的所有元素分成两个序列 A 和 B，并按照绝对值分别将 A 和 B 中的元素从小到大进行排序。使得 $\forall i<j$，都有 $a_i\in A\wedge a_j\in A\wedge |a_i|<|a_j|$ 和 $b_i\in B\wedge b_j\in B\wedge |b_i|<|b_j|$，并且 A 和 B 的元素个数之和为 m，即 $|A|+|B|=m$。令 $|A|=P$，$|B|=Q$。

2）在 A 和 B 中分别取若干个（总数不超过 k）绝对值较大的值：

①若 $P-\lceil k/2\rceil>0$，取 A 中前 $P-\lceil k/2\rceil$ 项并求和，即 $E_+=\sum_{i=1}^{P-\lceil k/2\rceil}a_i$；令 A 中前 $P-\lceil k/2\rceil$ 项值为 0；令 A 中余下的项为 $a_i=a_i+\alpha E_+$。

②若 $Q-\lfloor k/2\rfloor>0$，取 B 中前 $Q-\lfloor k/2\rfloor$ 项并求和，即 $E_-=\sum_{i=1}^{Q-\lfloor k/2\rfloor}b_i$；令 B 中前 $Q-\lfloor k/2\rfloor$ 项的值为 0；令 B 中余下的项为 $b_i=b_i+\alpha E_-$。

在上述步骤 2）中，若 $P-\lceil k/2\rceil>0$，则正数的个数超过 k 的一半，仅保留 $\lceil k/2\rceil$ 个正数，令绝对值小的正数为 0，并将这些小正数的和乘以一定的系数 α 加到保留的 $\lceil k/2\rceil$ 个正数上；若 $P-\lceil k/2\rceil\leqslant 0$，则正数的个数小于 k 的一半，不对这些正数执行任何操作。对 B 中各项的操作与对 A 中的元素的操作相同。

经过上述稀疏化后，全局调整函数 g 对 h 中各个元素的取值进行了调整，仅保留了不超过 k 个非 0 元素。在稀疏化过程中，并没有将绝对值小的元素的值舍去，而是将其转到了绝对值大的元素上。图 2 为一个隐藏层向量稀疏化过程的具体实例，其中 $m=5$，$k=2$。

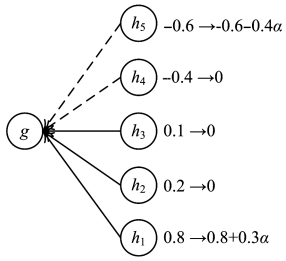

图 2　隐藏层向量稀疏化示意图

Fig. 2　Schematic diagram of vector sparsity in latent layer

将经过全局调整函数 g 对 h 稀疏化后得到的向量记为 h'，则有 $h'=g(h)$。

在得到 h' 后，继续应用全连接神经网络对 h' 进行重建，得到 o，计算公式为：

$$o=\tanh(W^{\mathrm{T}}h'+b') \tag{3}$$

其中，$\tanh(\cdot)$ 仍为双曲正切函数，W^{T} 为 W 的转秩，b' 为输出层向量 o 的偏差常量。

输入文本的向量化：在应用上述提出的自编码器之前，首先需要对文档中的文本进行向量化，将其转化为自编码器的输入向量 x。下面对 x 的含义进行说明。$x\in\mathbb{R}^d$，d 为语料库中单词的个数，x 中各个元素的定义如下：

$$x_i=\frac{\log(1+n_i)}{\max\limits_{j\in V}\log(1+n_j)},i\in V \tag{4}$$

其中，V 是语料库中的单词集合，$|V|=d$，n_j 为单词 j 在文档中出现的次数。

重建输入向量的目标函数：给定输入向量 x，得到重建后的输出向量 x'，采用交叉熵函数作为重建函数的损失函数。其计算公式为：

$$l(x,x')=-\sum_{i\in V}x_i\log(x_i')+(1-x_i)\log(1-x_i') \tag{5}$$

通过所提自编码器对文档中的单词进行嵌入式向量化表示后，采用 DBN 神经网络对文档进行分类。DBN 网络的详细结构与算法请参考文献[14]。

3　实验分析

3.1　实验数据

实验采用公开的 20 Newsgroups（20news）数据集[15]。该

数据集包括 18846 篇文档，这些文档被划分在 20 个不同的新闻组中，且每个文档只属于一个新闻组。数据集中的训练数据、测试数据和验证数据分别有 11314 篇、7532 篇和 1000 篇，整个数据集包含了 2000 个常用单词。

3.2 对比算法

将本文提出的改进自编码器分类算法记为 IAE，并与经典的词嵌入、文本分类算法进行对比。采用的对比算法有自动编码器（AE）[7]，Word2Vec[5]，Doc2Vec[6]，LDA[3] 和 DBN[14]。在这些算法中，AE，Word2Vec 和 Doc2Vec 3 种算法用于提取单词的嵌入式向量特征，并在此基础上采用单层全连接神经网络进行文本分类；DBN 采用多层自编码器进行文本特征向量的提取，然后采用单层的全连接网络进行文本分类。

3.3 实验结果

首先，对比 AE，IAE 和 Word2Vec 3 种算法的词嵌入效果。采用 3 种算法对数据集中的单词进行向量化后，随机选取 weapon，law，compute 和 space 4 个单词，然后按照相似性选取 5 个相似的单词，结果如表 1 所列。从表中可以看出，IAE 算法提取的结果具有很好的效果，其选出的单词与候选单词具有很近的语义[16-17]。

表 1 算法嵌入式向量中的近似单词对比

Table 1 Comparison of approximate words in embedded vector of algorithm

算法	weapon	law	compute	space
AE	effort	made	inform	study
	muslim	live	run	data
	sort	give	program	answer
	america	power	base	origin
	escape	call	author	unit
IAE	arm	citizen	scienc	launch
	crime	constitute	dept	orbit
	gun	court	cs	mission
	firearm	feder	math	shuttl
	handgun	govern	univ	flight
Word2Vec	assault	court	engin	launth
	militia	prohibit	colleg	jpl
	possess	ban	umich	nasa
	automat	sentence	subject	moon
	gun	legitim	perform	govern

其次，对比了 AE，IAE 和 LDA 3 种算法在进行文本分类时的均方余弦偏差（Mean Squared Cosine Deviation，MSCD）。令 m 为话题的总数，v_i 和 v_j 分别表示属于话题 i 和话题 j 的概率，$\cos(v_i,v_j)=\dfrac{v_i^{\mathrm{T}}v_j}{\|v_i\|\cdot\|v_j\|}$，则 MSCD 的计算公式为：

$$MSCD=\sqrt{\frac{2}{m\cdot(m-1)}\sum_{i,j>i}\cos^2(v_i,v_j)} \tag{6}$$

$MSCD\in[0,1]$，$MSCD$ 越大，表明两个话题越相似。图 3 为算法的 $MSCD$ 对比图。在 IAE 算法中，分别令参数 α 为 1 和 5.35。从图中可以看出，当 $N=20$ 时，IAE 算法的 $MSCD$ 略优于 LDA 算法，这两种算法的 $MSCD$ 明显优于 AE 算法；当 $N=40$ 和 $N=80$ 时，IAE 算法和 LDA 算法的 $MSCD$ 也明显优于 AE 算法，并且当 $\alpha=5.35$ 时，IAE 算法的 $MSCD$ 优于 LDA 算法。

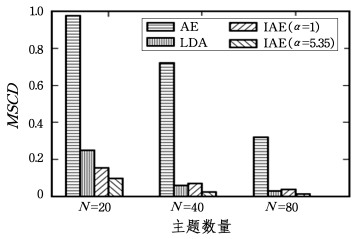

图 3 算法的 $MSCD$ 对比

Fig. 3 Comparison of $MSCD$ of algorithms

接着，对比 IAE，Word2Vec，Doc2Vec，LDA 和 DBN 5 种算法的分类准确性，对比结果如图 4 所示。从图中可以看出，IAE 算法的分类准确率明显优于其他算法。

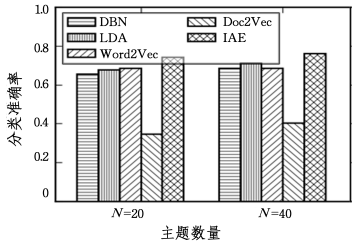

图 4 算法的分类准确率对比

Fig. 4 Comparison of classification accuracy of algorithms

最后，通过对 IAE 算法中各个参数进行调整，使得算法取得最好的性能。通过调整数据集中主题的数量，来观察 IAE 算法的分类准确性。从图 5 可以看出，随着主题数量的不断增加，IAE 算法的分类准确率首先增加，当达到 1024 时开始下降。通过调整嵌入式向量的维度 k 来观察 IAE 算法的分类准确性，实验结果如图 6 所示。从图 6 中可以看出，随着 k 的变化，IAE 算法的分类准确性也发生变化，当 $k=32$ 时，其分类准确率最高。通过调整 IAE 算法的全局调整函数 g 中参数 α 的取值来观察 IAE 算法的分类准确性，结果如图 7 所示。从图 7 中可以看出，当 α 的取值大概处于 6～9 之间时，IAE 算法的分类准确率最高。

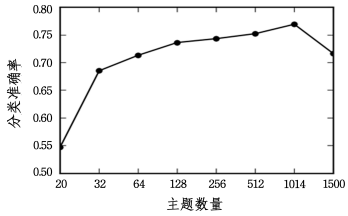

图 5 IAE 算法的分类准确率与主题数量之间的关系

Fig. 5 Relationship between classification accuracy and number of topics of IAE algorithm

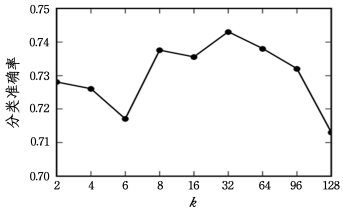

图 6 IAE 算法的分类准确率与特征向量维度 k 之间的关系

Fig. 6 Relationship between classification accuracy and dimension of feature vector k of IAE algorithm

（下转第 240 页）