

融合稀疏因子的情感分析堆叠降噪自编码器模型

蒋宗礼 王一大

(北京工业大学信息学部 北京 100124)

摘要 基于深度学习的特征抽取是目前数据降维问题的研究热点,堆叠自编码器作为一种较为常用的模型,无法对混有噪声及较稀疏的数据进行良好的特征表达。面向微博情感分析,通过在堆叠降噪自编码器的各隐藏层中加入稀疏因子,来解决样本数据所含噪声和稀疏性对特征抽取的影响。使用 COAE 评测数据集进行的情感分析实验表明所提模型分类的准确率和召回率都有所提高。

关键词 深度学习,堆叠降噪自编码器,稀疏因子,情感分析

中图分类号 TP183 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.12.041

Sentimental Analysis Stacked Denoising Auto-encoder with Sparse Factor

JIANG Zong-li WANG Yi-da

(Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China)

Abstract Feature extraction based on deep learning is a hot research topic in data dimensionality reduction now. In deep learning, stacked auto-encoder is commonly used. The encoder just simply learns the features of sample and can't get a good feature expression for the data which are mixed with noise and sparsity. Sparse factor is added in each hidden layer of the stacked denoising auto-encoder to solve the problem of feature extraction about data with noise and sparsity in this paper. The sentimental analysis experiments on COAE data set show that the precision and recall ratio are improved.

Keywords Deep learning, Stacked denoising auto-encoder, Sparse factor, Sentimental analysis

1 引言

深度学习是一类多层神经网络学习算法,近年来在语音识别、计算机视觉等领域取得了突破性进展,并引起了广泛关注^[1-3]。它通过组合低层特征形成更加抽象的高层表示,以发现数据的分布特征^[4]。这一概念最早由多伦多大学的 G. E. Hinton 等^[5]于 2006 年提出。本文选用的基础模型是由 Vincent 提出的堆叠降噪自编码器(Stacked Denoising AutoEncoder, SDAE)^[6],该模型解决了由数据中的噪声带来的影响,增强了算法的鲁棒性。研究表明,在多数情况下,SDAE 优于深度置信网络(Deep Belief Network, DBN)^[7]。

随着社交媒体的迅猛发展,越来越多的人通过微博等在网发表评论,这些评论褒贬不一。情感分析旨在实现评论内容情感倾向的自动分析,是自然语言处理领域中的研究热点^[8],同时由此衍生出的情感分类也具有很大影响^[9]。

传统情感分类方法主要依靠人工标注及情感词典,然而网络微博信息量庞大,进行人工标注费时、费力。因此,本文先使用深度学习对微博信息进行特征提取,再进行情感分类。文献^[10]和文献^[11]分别使用卷积神经网络对短文本信息进

行情感分类研究,其中文献^[11]针对中文情况,将词级别的向量转换为字级别的向量,使准确率有所提高。文献^[12]将堆叠降噪自编码器应用于亚马逊商品评论信息的情感分类研究中。文献^[13]使用递归自编码器进行情感分类研究,同时加入了情感极性转移。上述方法都取得了一定的效果,但都未考虑短文本数据本身所固有的稀疏性问题,即在数据向量中零元素远远多于非零元素的问题。

为防止微博数据中噪声数据对分类器的影响,选用堆叠降噪自编码器作为基础模型;同时为解决短文本稀疏性问题,在堆叠降噪自编码器的各隐藏层中加入稀疏因子,形成融合稀疏因子的堆叠降噪自编码器(Stacked Sparse Denoising AutoEncoder, SSDAE)模型,以提高模型的泛化能力,既保证了堆叠稀疏自编码器的稀疏性限制,又保证了堆叠降噪自编码器抗噪声的鲁棒性,从而提高了其特征抽取能力和情感分类的准确率。

2 堆叠降噪自编码器模型

Vincent 于 2008 年提出降噪自编码器模型,其结构如图 1 所示。该编码器先将噪声信息随机加在输入向量 x 上,得

到稿日期:2016-11-02 返修日期:2017-02-03

蒋宗礼(1956-),男,教授,CCF 会员,主要研究方向为网络信息搜索与处理;王一大(1991-),男,硕士生,CCF 学生会员,主要研究方向为机器学习。

到向量 \tilde{x} ;再对 \tilde{x} 进行编码,形成隐藏层的向量 y ;然后对 y 进行解码,形成重构后的信息 z ,通过最小化 x 与 z 的误差完成训练。

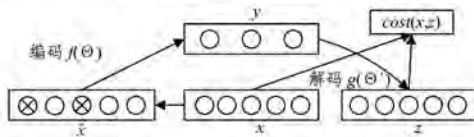


图1 降噪自编码器结构图

其实现方案如下:

假设输入的样本集合为 $X = \{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}$, 为表示简洁, 用 x 表示 $x^{(i)}$, 其中 $x = \{x_1, x_2, \dots, x_d\}$ 为输入向量, $y = \{y_1, y_2, \dots, y_h\}$ 为对应的隐藏层向量, $z = \{z_1, z_2, \dots, z_d\}$ 为对应的输出层向量。同时规定网络中各神经元激活函数为 sigmoid 函数。

$$f(x) = \text{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \quad (1)$$

输入向量通过 masking noise 随机映射成输入层的“受破坏”向量 \tilde{x} :

$$x \xrightarrow{\text{masking noise}} \tilde{x} \quad (2)$$

输入层到隐藏层的编码操作为:

$$y = f(W_y \tilde{x} + b_y) = \text{sigmoid}(W_y \tilde{x} + b_y) \quad (3)$$

隐藏层到输出层的解码操作为:

$$z = f(W_z y + b_z) = \text{sigmoid}(W_z y + b_z) \quad (4)$$

其中, W_y, W_z 分别是输入-隐层和隐层-输出的连接权值矩阵; b_y, b_z 分别是隐藏层、输出层神经元的偏置向量。

为使解码后的值接近输入向量, 设最小化重构误差目标函数为:

$$\text{idealreconstruction} = \arg \min_{W, b_y, b_z} [J(x, z)] \quad (5)$$

其中, $J(x, z)$ 为输入与输出的误差函数, 一般采用交叉熵损失函数:

$$J(x, z) = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^d [x_{ik} \log(z_{ik}) + (1 - x_{ik}) \log(1 - z_{ik})] \quad (6)$$

使用梯度下降法, 通过迭代修正网络神经元之间的连接权值矩阵及偏置向量求其极小值(其中 η 代表学习率):

$$W = W - \eta \frac{\partial J(x, z)}{\partial W} \quad (7)$$

$$b_y = b_y - \eta \frac{\partial J(x, z)}{\partial b_y} \quad (8)$$

$$b_z = b_z - \eta \frac{\partial J(x, z)}{\partial b_z} \quad (9)$$

多个降噪自编码器堆叠在一起, 将前一个隐藏层的输出作为后一个降噪自编码器的输入, 进行预训练; 训练完成后对整个网络进行微调, 即构建堆叠降噪自编码器。

堆叠降噪自编码器较好地解决了噪声问题。但是, 如果数据本身比较稀疏, 将会降低模型的泛化能力。

3 融合稀疏因子的堆叠降噪自编码器模型

微博口语话现象严重, 很多短文本的部分数据缺失, 特别

是当选取的特征向量维度较大时, 数据稀疏性问题更加严重。本文通过在堆叠降噪自编码器各隐藏层中加入稀疏因子来提高模型的泛化能力。

3.1 稀疏因子

稀疏因子的作用类似于正则化, 即防止模型陷入过拟合, 同时防止在计算过程中输入数据的稀疏性被破坏。因此, 需要使隐藏层的神经元大部分处于“抑制”状态, 少部分处于“激活”状态。

为使神经元在大部分时间处于“抑制”状态, 本文引入平均激活度, 用 $f_j(x_i)$ 表示输入层第 i 个节点到隐藏层第 j 个节点的输出值, 隐藏层第 j 个神经元的平均激活度为:

$$\hat{\rho}_j = \frac{1}{d} \sum_{i=1}^d f_j(x_i) = \frac{1}{d} \sum_{i=1}^d \text{sigmoid}(w_{ji} * x_i + b_j) \quad (10)$$

为使 $\hat{\rho}_j = \rho$ (其中 ρ 为稀疏性参数), 令 ρ 尽可能小, 以实现稀疏限制。此时, 隐藏层神经元的激活值分布必须大部分接近于 0。为此, 以额外的稀疏误差函数作为衡量标准:

$$J_{\text{sparse}}(W, b) = KL(\rho | \hat{\rho}_j) = \beta \sum_{j=1}^h (\rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j}) \quad (11)$$

其中, h 为隐藏层神经元个数, β 为稀疏惩罚因子, $KL(\rho | \hat{\rho}_j)$ 是分别以 ρ 为均值和以 $\hat{\rho}_j$ 为均值的两个伯努利随机变量之间的相对熵。通过稀疏误差函数来限制稀疏表达。

3.2 融合稀疏因子的堆叠降噪自编码器

融合稀疏因子的堆叠降噪自编码器模型的结构如图 2 所示, 通过在编码过程中加入稀疏因子来实现稀疏性限制, 形成隐藏层的“稀疏表达”(图 2 中用 y 表示); 然后使用激活函数来对隐藏层各神经元进行解码计算, 最后使用梯度下降法修正网络神经元之间的连接权值, 完成网络训练。

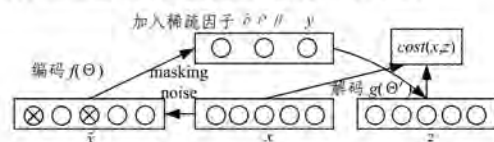


图2 融合稀疏因子的降噪自编码器结构图

各隐藏层计算稀疏项 $J_{\text{sparse}}(W, b)$ 时, 加入稀疏性参数 ρ 、稀疏惩罚因子 β 和各隐藏层节点的平均激活度 $\hat{\rho}$, 其中 $\hat{\rho} = \{\hat{\rho}_1, \dots, \hat{\rho}_j, \dots, \hat{\rho}_h\}$, h 为隐藏层神经元个数。模型整体的重构误差函数由两部分构成: 1) 输入与输出的误差函数 $J(x, z)$; 2) 稀疏误差函数 $J_{\text{sparse}}(x, z; W, b)$, 即:

$$J(x, z; W, b) = J(x, z) + J_{\text{sparse}}(W, b) \quad (12)$$

使用梯度下降法, 迭代更新网络神经元之间的连接权值矩阵及偏置向量(其中 η 代表学习率):

$$\Delta W = \Delta W + \frac{\partial J(x, z; W, b)}{\partial W} \quad (13)$$

$$\Delta b_y = \Delta b_y + \frac{\partial J(x, z; W, b)}{\partial b_y} \quad (14)$$

$$\Delta b_z = \Delta b_z + \frac{\partial J(x, z; W, b)}{\partial b_z} \quad (15)$$

$$W = W - \eta \Delta W \quad (16)$$

$$b_y = b_y - \eta \Delta b_y \quad (17)$$

$$b_c = b_c - \eta \Delta b_c \quad (18)$$

融合稀疏因子的降噪自编码器模型训练算法如下。

输入:训练样本 $x = \{x_1, x_2, \dots, x_d\}$

输出: $y = \{y_1, y_2, \dots, y_n\}; \Delta W = 0; \Delta b_y = 0; \Delta b_c = 0$

随机初始化 W, b_y, b_c ;

1. 使用式(2)生成 \bar{x} ;

2. 使用式(10)计算 $\hat{\rho}_1, \dots, \hat{\rho}_j, \dots, \hat{\rho}_k$;

3. 使用式(3)进行编码计算,计算隐藏层神经元 z ;

4. 使用式(4)进行解码计算,计算输出层神经元 y ;

5. while 不收敛 do;

 使用式(6)、式(11)、式(12)计算 $J(x, z; W, b)$;

 使用式(13)–式(15)计算 $\Delta W, \Delta b_y, \Delta b_c$;

 使用式(16)–式(18)更新 W, b_y, b_c ;

end while;

以上描述的是单层网络模型,为构成堆叠网络,只需将前一个单层网络的结果 y 传递给下一个单层网络(作为输入向量),当最后一个单层网络预训练结束时,再从后向前进行整体网络微调。

该模型一方面保留了堆叠降噪自编码器对噪声数据的鲁棒性,另一方面通过稀疏因子的稀疏性限制保证了特征的稀疏表达,提高了模型的泛化能力。

4 微博情感分析应用实验

为测试融合稀疏因子的堆叠降噪自编码器模型的效果,本文将应用于中文微博情感分类。实验数据来自 COAE2015 微博数据测评集,从中随机抽取 700 条文本作为训练集,600 条文本作为测试集。

4.1 实验系统设计

由于融合稀疏因子的堆叠降噪自编码器模型仅用于提取输入数据的特征信息且并未进行分类,因此需要将提取后的信息输入 Softmax 分类器中完成分类任务。

为验证改进后的模型的效果,设计了以下对比实验。

实验 1 堆叠降噪自编码器(SDAE)与融合稀疏因子的堆叠降噪自编码器(SSDAE)的对比,以验证融入稀疏因子后模型的效果。

实验 2 Softmax 模型与 SSDAE+Softmax 模型的对比,以验证浅层学习模型与深度学习模型的效果。

实验 3 深度置信网络(DBN)与 SSDAE 的对比,以验证 SSDAE 与深度置信网络模型的效果。

另外为探索稀疏因子的选取,针对不同对象进行了 4 组稀疏操作实验。

以 SSDAE 微博情感分类系统为例,其整体架构如图 3 所示。

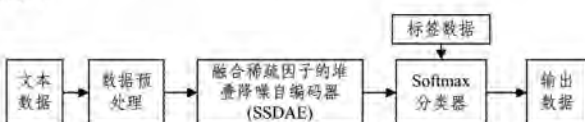


图 3 微博情感分类系统的整体架构图

实验过程:

1)以非放回方式随机从 COAE2015 数据集中抽取 700 条文本作为训练集,600 条文本作为测试集;

2)对文本进行分词,去停用词,使用 $tf \cdot idf$ 、卡方检验方法初步抽取特征值根据初步抽取的特征值分别生成训练数据集和测试数据集的向量;

3)将训练集的向量输入融合稀疏因子的堆叠降噪自编码器模型中进行预训练;

4)将模型提取出的特征信息输入 Softmax 分类器中进行分类计算,同时将标签数据输入 Softmax 分类器中进行监督训练;

5)当 Softmax 分类器计算结束后,将网络从后至前进行整体微调。

以上架构是一个通用框架,通过将 SSDAE 分别替换成 SDAE,DBN 和 NULL 来完成关于 SDAE,DBN 和 Softmax 微博情感分类的实验;在针对不同对象进行稀疏处理的 4 组对比实验中,只需改动 SSDAE 模块中的稀疏因子计算函数即可。

4.2 实验结果与分析

本实验使用传统分类性能评价指标:准确率(accuracy)、精确率(precision)、召回率(recall)、 F -测度值。其计算公式如下:

$$\text{准确率}(A) = \frac{\text{被正确分类的文本个数}}{\text{文本总个数}} \quad (19)$$

$$\text{精确率}(P) = \frac{\text{被正确分类的文本个数}}{\text{实际分到该类的文本个数}} \quad (20)$$

$$\text{召回率}(R) = \frac{\text{正确分到该类的文本个数}}{\text{实际应分到该类的文本个数}} \quad (21)$$

$$F = \frac{2 * P * R}{(P + R)} \quad (22)$$

(1)不同学习模型对分类结果的影响

表 1 列出了各学习模型在全部实验数据上的结果,其中 A 表示准确率(accuracy), P 表示精确率(precision), R 表示召回率(recall), F 表示 F -测度值。

通过表 1 可以得出以下信息:

1)SDAE+Softmax 模型与 SSDAE+Softmax 模型相比,在初始向量维度为 3276 时,SDAE+Softmax 的准确率为 59.5%,SSDAE+Softmax 的准确率为 64.5%,整体准确率提高了 5%;不同类别的 F -测度值也有所提升,中性的 F -测度值由 72.84%提高到 79.15%,褒义的 F -测度值由 44.09%提高到 51.41%,贬义的 F -测度值由 41.16%提高到 46.85%。可以看出融合稀疏因子的堆叠降噪自编码器在堆叠降噪自编码器的基础上使分类效果有所提升。

2)Softmax 模型与 SSDAE+Softmax 模型相比,在初始向量维度为 3276 时,Softmax 的准确率为 58%,SSDAE+Softmax 的准确率为 64.5%,整体准确率提高了 6.5%;不同类别的 F -测度值也有所提升,中性的 F -测度值由 66.85%提高到 79.15%,褒义的 F -测度值由 42.86%提高到 51.41%,贬义的 F -测度值由 41.63%提高到 46.85%,这表明融合稀

疏因子的堆叠降噪自编码器相对于浅层 Softmax 模型、SS-DAE 模型,能更紧凑高效地表达显示问题中的非线性特征,对于微博情感分类具有更好的分类效果。

3)DBN+Softmax 模型与 SSDAE+Softmax 模型相比,在初始向量维度为 3276 时,DBN+Softmax 的准确率为 59.5%,SSDAE+Softmax 的准确率为 64.5%,整体准确率提

高了 5%;同时不同类别的 F -测度值也有所提升,中性的 F -测度值由 73.53%提高到 79.15%,褒义的 F -测度值由 43.75%提高到 51.41%,贬义的 F -测度值由 42.38%提高到 46.85%,这表明在使用相同分类器的情况下,融合稀疏因子的堆叠降噪自编码器相对于深度置信网络模型,在微博情感分类领域中对特征的提取能力更好。

表 1 不同学习模型的实验结果/%

初始向量 维度	模型	褒义			中性			贬义			A
		R	P	F	R	P	F	R	P	F	
3276	Softmax	48.21	38.57	42.86	63.16	70.99	66.85	40.30	43.04	41.63	58.00
	DBN+Softmax	43.44	44.06	43.75	88.29	63.00	73.53	39.72	45.43	42.38	59.50
	SDAE+Softmax	45.07	43.15	44.09	90.09	61.14	72.84	40.56	41.78	41.16	59.50
	SSDAE+Softmax	47.34	56.25	51.41	97.30	66.70	79.15	43.31	51.02	46.85	64.50
3173	Softmax	66.15	46.24	54.43	60.31	58.97	59.63	43.00	48.14	45.43	53.00
	DBN+Softmax	63.92	50.83	56.63	85.00	62.80	72.23	42.43	45.17	43.76	55.00
	SDAE+Softmax	61.62	51.03	55.83	84.00	61.17	70.79	45.21	47.13	46.15	55.00
	SSDAE+Softmax	61.72	52.00	56.44	87.50	63.47	73.57	43.20	50.57	46.60	60.00
2245	Softmax	50.00	40.77	44.92	43.48	60.82	50.71	41.90	44.52	43.17	58.00
	DBN+Softmax	48.21	39.57	43.47	65.22	59.73	62.35	37.45	43.29	40.16	55.00
	SDAE+Softmax	49.86	43.81	46.64	75.65	55.77	64.21	42.56	45.12	43.80	58.50
	SSDAE+Softmax	53.57	46.67	49.88	100.00	62.50	76.92	43.33	45.00	44.15	65.00
1428	Softmax	74.63	54.35	62.90	64.08	58.24	61.02	42.86	45.88	44.32	57.00
	DBN+Softmax	66.42	45.83	54.24	66.33	50.39	57.27	43.14	37.66	40.21	52.00
	SDAE+Softmax	73.73	55.37	63.24	69.39	57.37	62.81	42.86	47.10	44.88	57.50
	SSDAE+Softmax	72.69	58.55	64.86	70.31	63.31	66.63	47.14	45.30	46.20	62.50

综上所述,在不同维度的微博信息中,通过使用稀疏因子对模型进行稀疏性限制,能使模型抽取出的特征更好地表达输入数据的结构特征。融合稀疏因子的堆叠降噪自编码器模型相对于堆叠降噪自编码器的泛化能力有所提高,并使其在情感分类问题中的分类效果优于其他 3 种模型。

(2)不同对象进行稀疏处理对分类结果的影响

实验中发现,融合稀疏因子的堆叠降噪自编码器在情感分类实验的训练阶段比较慢,其原因是在计算稀疏因子阶段

进行 $Wx+b$ 操作消耗了大量的时间。因此,实验中使用控制变量法对稀疏因子对象的选取做了对比实验,结果如表 2 所列。其中 A 表示准确率(accuracy),P 表示精确率(precision),R 表示召回率(recall),F 表示 F-测度值,T 表示训练时间。SSDAE 表示在稀疏因子的计算中只针对输入进行稀疏操作;SSDAE-W 表示在稀疏因子的计算中针对隐藏层的网络输入进行稀疏操作;SSDAE-Wb 表示在稀疏因子的计算中针对加入偏移量后隐藏层的网络输入进行稀疏操作。

表 2 不同对象进行稀疏处理的结果

初始向量 维度	模型	褒义/%			中性/%			贬义/%			A/%	T/天
		R	P	F	R	P	F	R	P	F		
3276	SSDAE-Wb+Softmax	47.34	56.25	51.41	97.30	66.70	79.15	43.31	51.02	46.85	64.50	4.00
	SSDAE-W+Softmax	48.93	53.45	51.09	96.20	63.67	76.63	44.20	43.50	43.85	62.50	1.70
	SSDAE+Softmax	47.86	52.62	50.13	95.39	63.10	75.96	43.50	44.10	43.80	62.00	1.55
3173	SSDAE-Wb+Softmax	61.72	52.00	56.44	87.50	63.47	73.57	43.20	50.57	46.60	60.00	4.00
	SSDAE-W+Softmax	60.77	51.28	55.62	86.00	63.42	73.00	43.12	50.50	46.45	58.00	1.65
	SSDAE+Softmax	60.23	51.24	55.37	84.00	63.50	72.33	42.50	50.30	46.43	56.50	1.50
2245	SSDAE-Wb+Softmax	53.57	46.67	49.88	100.00	62.50	76.92	43.33	45.00	44.15	65.00	3.60
	SSDAE-W+Softmax	53.47	45.86	49.37	100.00	61.38	76.07	43.19	44.48	43.83	63.50	1.40
	SSDAE+Softmax	31.79	100.00	49.16	100.00	60.79	75.61	43.20	43.69	43.44	63.00	1.30
1428	SSDAE-Wb+Softmax	72.69	58.55	64.86	70.31	63.31	66.63	47.14	45.30	46.20	62.50	3.30
	SSDAE-W+Softmax	72.24	56.94	63.68	70.41	60.50	65.08	46.42	44.05	45.20	60.00	1.30
	SSDAE+Softmax	71.22	57.00	63.32	69.29	63.00	66.00	46.11	44.92	45.51	58.50	1.20

通过表 2 可以得出以下信息:

1)SSDAE-Wb+Softmax 模型与 SSDAE-W+Softmax 模型以及 SSDAE+Softmax 模型的实验结果相比,当初始向量维度为 3276 时,SSDAE-Wb+Softmax 的准确率为 64.5%,SSDAE-W+Softmax 的准确率为 62.5%,SSDAE+Softmax 的准确率为 62%,SSDAE-Wb+Softmax 比 SSDAE-W+Softmax 提高了 2%,而 SSDAE-W+Softmax 比 SSDAE+Softmax 提高了 0.5%;SSDAE-Wb+Softmax 的训练时间开销为

4 天,SSDAE-W+Softmax 的训练时间开销为 1.7 天,SSDAE+Softmax 的训练时间开销为 1.55 天,这表明在稀疏因子的计算中,针对加入偏移量后对隐藏层的网络输入进行稀疏操作所得到模型分类效果最好,但其训练时间开销最大。

2)SSDAE-Wb+Softmax 模型与 SSDAE-W+Softmax 模型相比,当初始向量维度为 3276 时,SSDAE-Wb+Softmax 的准确率为 64.5%,SSDAE-W+Softmax 的准确率为

62.5%,准确率相对减少了 2%;SSDAE-Wb+Softmax 的训练时间开销为 4 天,SSDAE-W+Softmax 的训练时间开销为 1.7 天,这表明若在稀疏因子的计算中仅针对隐藏层的网络输入进行稀疏操作所得到的模型,其训练效果虽然不如加入偏移量后对隐藏层的网络输入进行稀疏操作所得到的模型,但在训练时间的开销上却得到了显著提高,训练时间缩短为原来的 1/2。

3)SSDAE-W+Softmax 模型与 SSDAE+Softmax 模型相比,当初始向量维度为 3276 时,SSDAE-W+Softmax 的准确率为 62.5%,SSDAE+Softmax 的准确率为 62%,准确率相对减少了 0.5%;SSDAE-W+Softmax 的训练时间开销为 1.7 天,SSDAE+Softmax 的训练时间开销为 1.55 天,这表明若在稀疏因子的计算中只针对输入进行稀疏操作,该模型在训练阶段的时间开销是最少的,但分类效果却不如之前两种对象进行稀疏操作后所得到的效果理想。

综上所述,若单纯考虑分类准确率,应针对加入偏移量后隐藏层的网络输入进行稀疏操作,这样所得到的模型分类效果最好;若同时考虑训练时间开销,并保证分类准确率在一定容忍度之内,可针对隐藏层的网络输入进行稀疏操作。

此外,通过表 1 中 SDAE+Softmax 模型与表 2 中 SSDAE+Softmax 模型的实验结果对比可知,当初始向量维度为 3276 时,即使只针对输入进行稀疏操作,所得到的模型也比原始模型的准确率提高了 2.5%,在其他初始向量维度下同样提高了 1%~4.5%,这表明加入稀疏因子有效保证了原始输入数据的稀疏表达,使模型的泛化能力得到了提升。

结束语 融合稀疏因子的堆叠降噪自编码器模型在堆叠降噪自编码器模型的基础上,通过融合稀疏因子保证了稀疏样本的稀疏表达,解决了样本数据在含有噪声且稀疏情况下的特征抽取问题,并提高了模型的泛化能力。分类实验表明该模型优于单纯使用堆叠降噪自编码器的模型。另外,选取不同稀疏因子时训练的时间开销不同,可以通过实验发现时间开销不算太大,但效果较好的稀疏操作。

由于深度学习需要大数据支持从而导致训练时间开销较大,因此进一步工作将深入研究融合稀疏因子的堆叠降噪自编码器的并行化方法。

参 考 文 献

- [1] BENGIO Y, DELALLEAU O. On the expressive power of deep architectures[C]// International Conference on Discovery Science. Springer-Verlag, 2011: 1.
- [2] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc, 2012: 1097-1105.
- [3] DAHL G E, YU D, DENG L, et al. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition[J]. IEEE Transactions on Audio Speech & Language Processing, 2012, 20(1): 30-42.
- [4] KARPATHY A, TODERICI G, SHETTY S, et al. Large-Scale Video Classification with Convolutional Neural Networks[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2014: 1725-1732.
- [5] HINTON G E, OSINDERO S, TEH Y W. A fast learning algorithm for deep belief nets[J]. Neural Computation, 2006, 18(7): 1527-1554.
- [6] VINCENT P, LAROCHELLE H, BENGIO Y, et al. Extracting and composing robust features with denoising autoencoders[C]// International Conference on Machine Learning. DBLP, 2008: 1096-1103.
- [7] CAO L L, HUANG W B, SUN F C. Building feature space of extreme learning machine with sparse denoising stacked-autoencoder[J]. Neurocomputing, 2016, 174(PA): 60-71.
- [8] RETRIEVAL F T I. Opinion Mining and Sentiment Analysis [J]. Foundations & Trends in Information Retrieval, 2008, 2(1): 459-526.
- [9] PANG B, LEE L, VAITHYANATHAN S. Thumbs up?: sentiment classification using machine learning techniques[C]// Conference on Empirical Methods in Natural Language Processing (Acl-02). Association for Computational Linguistics, 2002: 79-86.
- [10] STOJANOVSKI D, STREZOSKI G, MADJAROV G, et al. Twitter Sentiment Analysis Using Deep Convolutional Neural Network[C]// Hais. 2015: 726-737.
- [11] LIU F L, YANG L, ZHANG S W, et al. Convolutional Neural Networks for Chinese Micro-blog Sentiment Analysis [J]. Journal of Chinese Information Processing, 2015, 29(6): 159-165. (in Chinese)
刘龙飞, 杨亮, 张绍武, 等. 基于卷积神经网络的微博情感倾向性分析[J]. 中文信息学报, 2015, 29(6): 159-165.
- [12] GLOROT X, BORDES A, BENGIO Y. Domain adaptation for large-scale sentiment classification: A deep learning approach [C]// International Conference on Machine Learning. Omnipress, 2011: 513-520.
- [13] LIANG J, CHAI Y M, YUAN H B, et al. Deep Learning for Chinese Micro-blog Sentiment Analysis[J]. Journal of Chinese Information Processing, 2014, 28(5): 155-161. (in Chinese)
梁军, 柴玉梅, 原慧斌, 等. 基于深度学习的微博情感分析[J]. 中文信息学报, 2014, 28(5): 155-161.