文章编号：2096-1618(2016)增-0001-06

基于混合自动编码器的分类应用

蔡姣姣， 何 嘉
(成都信息工程大学计算机学院，四川 成都 610225)

摘要：自动编码器是一种无监督的深度学习模型，通过重构输入数据来提取数据特征。针对重构误差小但分类结果不一定好的问题，提出一种混合自动编码器模型。该方法在保证重构误差小的同时，使学习到的特征更有利于分类。实验结果表明，该方法对黑白和灰度图像的分类效果显著，准确率提高了1.5 % ~17 %。

关 键 词：深度学习；无监督学习；自动编码器；分类

0 引言

分类[1]是机器学习领域的关键问题之一。在简单的分类问题中，许多传统学习算法如决策树[2]、随机森林[3]、贝叶斯[4]、支持向量机[5]等算法，对数据具有较强的非线性拟合能力，能取得令人满意的分类效果。但对于现实生活中更多复杂的分类问题，传统学习算法不能精确拟合部分高维复杂函数[6]，必须先对数据进行特征提取，利用提取出的特征代替原始数据进行分类。Hinton 等[7]于 2006 年提出深度学习模型，展现了强大的特征学习能力，深度学习模型因此被广泛应用于分类问题的特征抽取阶段。

自动编码器(auto-encoder，AE)[8-11]是深度学习的主流网络模型之一。因结构简单直观，且在特征提取方面能取得良好的效果，在语音识别、图像识别、自然语言处理等领域具有广泛应用。自动编码器采用无监督学习方式，通过重构输入数据来提取特征，使提取的特征以更简单的形式表达原始信息，再把特征传给分类器进行分类。为了提高特征提取的效率，很多学者对自动编码器做出改进，深化自动编码器的研究。2007 年，Benjio 对自动编码器加入稀疏性限制[12]，提出稀疏自动编码器模型。该模型只选取最重要的信息，防止学习过多的噪声。2008 年，Vincent 通过向输入数据中添加噪声，提出降噪自动编码器[13]，增强了模型的抗噪性能。2010 年，Salah 通过限制升维和降维的过程，提出收缩自动编码器[14]的概念，将原始数据映射到高维空间，提高了模型对不同输入数据的鲁棒性。2011 年，Jonathan 将自动编码器与卷积神经网络结合，提出卷积自动编码器[15]，用于处理图像数据时分类效果显著提升。

自动编码器的无监督学习[16]方式倾向于学习利于重构的特征，没有其他指导，无法判断哪些是“好的”特征[17]。对于高维复杂函数，重构的数据含大量噪声，不利于后续的分类操作[18-19]。针对该问题，提出一种新的混合自动编码器模型(hybrid auto-encoder，HAE)。该模型通过引入 Softmax 分类器[20-21]，在重构原始数据的同时对数据进行类别监督，使模型提取出重构误差小并有利于分类的特征，提高分类准确率。

1 相关算法

1.1 自动编码器(AE)

自动编码器的结构如图 1 所示，由输入层(input)、隐含层(hidden)、输出层(output) 3 层组成。其中输入层神经元个数与输入数据的维数相同。隐含层是自动编码器的核心，表达的信息是网络从输入数据中学习到的特征。“+1”为神经元偏置项。输出层与输入层神经元个数一致，输出值由隐含层的特征重构得到。通过最小化重构输出值与输入层原始数据的误差，来更新整个网络的参数，使隐含层学习的特征作为输入数据更简单的表达。采用平方误差的形式衡量自动编码器的重构效果，单个样例的损失函数为

$$J(W,b;x,\hat{x}) = \frac{1}{2}||x-\hat{x}||^2 \qquad (1)$$

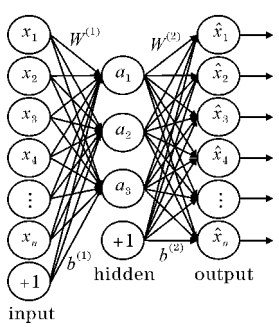


图 1 自动编码器

收稿日期：2016-02-19

其中,输入向量 $x=(x_1,x_2,\cdots,x_n)$,网络参数 $(W,b)=(W^{(1)},W^{(2)},b^{(1)},b^{(2)})$,$l$ 为网络的层数,$W^{(l)}$ 为第 l 层的连接参数、$b^{(l)}$ 为第 l 层的偏置项,$a^{(l)}$ 为第 l 层的激活值。输入层到隐含层的映射关系为

$$a^{(2)}=f(W^{(1)}x+b^{(1)}) \tag{2}$$

其中,f 为 sigmoid 激活函数,表达式为 $f(z)=1/(1+\exp(-z))$。$a^{(2)}$ 为隐含层学习到的特征,其值由网络的参数决定。对隐含层特征进行逆变换得到重构输出值

$$\hat{x}=a^{(3)}=f(W^{(2)}a^{(2)}+b^{(2)}) \tag{3}$$

自动编码器的目标是通过不断调节参数,使重构值与原始输入构成的损失函数 $J(W,b)$ 达到最小。

为防止出现参数过大,实际应用中将在损失函数中添加权重衰减项。同时,受稀疏编码理论的启发,对网络加入稀疏性约束。采用批量梯度下降法,由 m 个未带标签的样本 $(x^{(1)},x^{(2)},\cdots,x^{(i)},\cdots,x^{(m)})$ 组成训练集,总的损失函数为

$$J(W,b)=\frac{1}{m}\sum_{i=1}^{m}J(W,b;x^{(i)},\hat{x}^{(i)})+\frac{\lambda}{2}\sum_{l=1}^{n_l-1}(W^{(l)})^2+\beta\sum_{j=1}^{s^2}KL(\rho\,||\,\hat{\rho}_j) \tag{4}$$

其中,$J(W,b;x^{(i)},\hat{x}^{(i)})$ 表达式见式(1)。第二项为权重衰减项,λ 是该项的控制系数。第三项为稀疏性惩罚项,也称相对熵(KL divergence),具体表达式为

$$KL(\rho\,||\,\hat{\rho}_j)=\rho\log\frac{\rho}{\hat{\rho}_j}+(1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j} \tag{5}$$

ρ 是稀疏性参数,$\hat{\rho}_j$ 为隐含层第 j 个神经元的平均激活度(即隐含层的输出)

$$\hat{\rho}_j=\frac{1}{m}\sum_{i=1}^{m}a_j^{(2)}(x^{(i)}) \tag{6}$$

通过求导方式对公式(4)进行多次迭代来最小化损失值,当迭代次数超过预先设定次数或损失值达到某个给定的非常小的值时停止迭代,模型训练结束。给定任意的输入数据 x,利用训练得到的模型参数 (W,b) 计算隐含层的激活值 $a^{(2)}$。相比原始输入,$a^{(2)}$ 可能是一个更好的特征描述,将其取代 x 传入分类器中。

1.2 Softmax 回归

Softmax 回归模型因其简单有效,是深度学习中最常用的多分类器。自动编码器训练完成后,把提取的特征作为输入传给分类器进行分类。与直接输入原始数据相比,使用提取的特征将取得更好的分类效果。

假设有 m 个样本 $\{(x^{(1)},y^{(1)}),\ldots,(x^{(m)},y^{(m)})\}$,分为 k 个类别,当与自动编码器组合使用时,每一个输入 $x^{(i)}\in\Re^{n+1}$ 为自动编码器提取出的特征。对应类标记 $y^{(i)}\in\{1,2,\cdots,k\}$ 可以取 k 个不同的值,我们需要估算出每一个类别 j $(j=1,2,\cdots,k)$ 的概率 $p(y=j\,|\,x)$。Softmax 分类器的激活函数为

$$h_\theta(x)=\begin{bmatrix}p(y=1\,|\,x;\theta)\\p(y=2\,|\,x;\theta)\\\vdots\\p(y=k\,|\,x;\theta)\end{bmatrix}=\frac{1}{\sum_{i=1}^{k}e^{\theta_i^Tx}}\begin{bmatrix}e^{\theta_1^Tx}\\e^{\theta_2^Tx}\\\vdots\\e^{\theta_k^Tx}\end{bmatrix} \tag{7}$$

其中,$\theta\in\Re^{k\times(n+1)}$,$\theta_i\in\Re^{n+1}$ $(i=1,2,\cdots,k)$ 为模型的全部参数

$$\theta=\begin{bmatrix}\theta_1^T\\\theta_2^T\\\vdots\\\theta_k^T\end{bmatrix} \tag{8}$$

对模型加入权重衰减项进行优化,总体损失函数为

$$J(\theta)=-\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k}1\left\{y^{(i)}=j\right\}\log\frac{e^{\theta_j^Tx^{(i)}}}{\sum_{l=1}^{k}e^{\theta_l^Tx^{(i)}}}\right]+\frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{ij}^2 \tag{9}$$

其中,$1\{\cdot\}$ 是示性函数,$1\{True\}=1$,$1\{False\}=0$。

使用迭代的优化算法(例如梯度下降法,L-BFGS)更新参数,求导后的梯度公式为

$$\nabla_{\theta_j}J(\theta)=-\frac{1}{m}\sum_{i=1}^{m}\left[x^{(i)}(1\left\{y^{(i)}=j\right\}-p(y^{(i)}=j\,\middle|\,x^{(i)};\theta))\right]+\lambda\theta_j \tag{10}$$

其中,$\nabla_{\theta_j}J(\theta)$ 是一个向量,$\frac{\partial J(\theta)}{\partial\theta_{jl}}$ 是 $J(\theta)$ 对 θ_j 的第 l 个分量的偏导数。当使用梯度下降法时,参数的更新准则为:$\theta_j:=\theta_j-\alpha\nabla_{\theta_j}J(\theta)$ $(j=1,2,\cdots,k)$。通过最小化 $J(\theta)$ 训练模型,得到一个可用的 Softmax 分类器。

1.3 基于自动编码器的分类

由自动编码器和 Softmax 分类器组合,构成完整的分类模型,结构如图 2 所示,由两个阶段组成。第一阶段由自动编码器的输入层和隐含层构成,完成对输入数据的特征提取。第二阶段为 Softmax 分类器,其输入为自动编码器的隐含层,输出层的每个值为输入数据对应的各个类别概率值,取最大值为该输入的类标,即该模型对输入数据做出的分类结果。

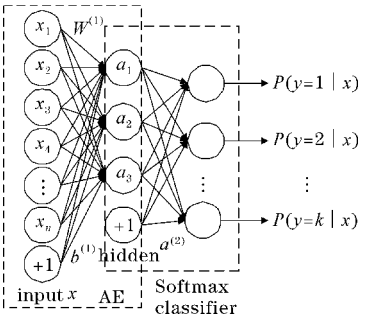


图2　自动编码器与 Softmax 回归的组合分类模型

自动编码器提取的特征实现了重构误差最小化的目标,但分类效果不一定好。改进后的混合自动编码器旨在提取利于分类的特征,从而提高第二阶段的分类准确率。

2　混合自动编码器(HAE)

从不同的角度看事物,往往可以有不同的表达形式。自动编码器提取的特征能够使重构误差达到最小,但分类效果不一定好。Softmax 分类器用于根据类别信息对学习数据之间的关系从而进行分类。

为提高自动编码器对类别的判断能力,提取出更有利于分类的特征,文中对自动编码器的结构和训练方式做出如下改进:在自动编码器的训练阶段,对输出层引入 Softmax 分类器,构成新的特征提取模型,称为混合自动编码器(hybrid auto-encoder)。混合自动编码器在重构原始数据的同时,对隐含层进行有监督的分类训练,结构如图 3 所示。

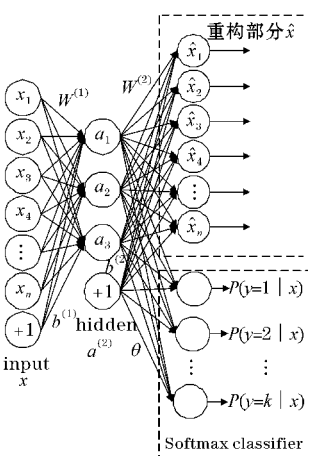


图3　混合自动编码器

对输出层添加一个分类器,总的损失函数也必须加上分类器部分的误差。新的损失函数表达式为

$$J = J(W,b) + \gamma J(\theta) \tag{11}$$

其中,$J(W,b)$ 见公式(4)定义,$J(\theta)$ 见公式(9)定义,γ 用于调节两项的权重。

假设有 m 个带标签的样本 $\{(x^{(1)},y^{(1)}),\cdots,(x^{(i)},y^{(i)}),\cdots,(x^{(m)},y^{(m)})\}$,$x^{(i)}$ 是第 i 个样本,$y^{(i)}$ 为对应输入的标签值。l 为网络的层数,$W^{(l)}$ 第 l 层的连接参数、$b^{(l)}$ 为第 l 层的偏置项,$a^{(l)}$ 为第 l 层的激活值。z^l 表示第 l 层神经元的输入值加权和,s_l 为第 l 层的神经元个数,输入层到隐含层和隐含层到输出层的激活函数均为 sigmoid 函数。输入层神经元的激活值为输入数据本身,对单个样本 x,令 $a^{(1)} = x$,混合自动编码器训练过程如下:

(1)前向传播计算各层激活值,输入层到输出层的重构部分计算公式为

$$\begin{cases} a^{(1)} = x \\ z^{(2)} = W^{(1)}a^{(1)} + b^{(1)} \\ a^{(2)} = f(z^{(2)}) \\ z^{(3)} = W^{(2)}a^{(2)} + b^{(2)} \\ \hat{x} = a^{(3)} = f(z^{(3)}) \end{cases} \tag{12}$$

输出层的 Sofmtax 分类器部分的激活函数参见公式(7)定义,只需把该函数的输入数据改为隐含层的激活值 $a^{(2)}$ 即可。

(2)误差反向传播。对输出层节点,采用梯度下降法对损失函数求导来更新网络参数。但隐含层的节点无法直接求导,因此以残差的形式将输出层的梯度误差进行反向传播。

对输出层,重构部分的节点残差 $\delta^{(3)}$ 为

$$\delta^{(3)} = \frac{\partial}{\partial z^{(3)}} \frac{1}{2} \| x - \hat{x} \|^2 = -(x - \hat{x}) \cdot f'(z^{(3)}) \tag{13}$$

Softmax 分类器部分的残差为 $I - P$ [21],其中 I 为类别标签向量,每个分量为输入数据对应的类别标签;P 为类别概率值 $p(y = j \mid x)$ 表示的条件概率向量。

对隐含层,残差为重构部分梯度值和分类器部分的梯度值之和

$$\delta^{(2)} = \left[((W^{(2)})^T \delta^{(3)}) + \beta(-\frac{\rho}{\hat{\rho}} + \frac{1-\rho}{1-\hat{\rho}}) + \gamma\theta^T(I - P) \right] \cdot f'(z^{(2)}) \tag{14}$$

(3)对所有 $l = 1,2$,利用残差计算每个参数的偏导数,Sofmtax 分类器部分神经元的参数偏导和更新参见 1.2 节相关公式,有

$$\begin{cases} \nabla_{W^{(l)}} J(W,b) = \delta^{(l+1)}(a^{(l)}) \\ \nabla_{b^{(l)}} J(W,b) = \delta^{(l+1)} \end{cases} \tag{15}$$

(4)更新所有参数

$$\begin{cases} W^{(l)} := W^{(l)} - \alpha[(\frac{1}{m}\nabla_{W^{(l)}} J(W,b)) + \lambda W^{(l)}] \\ b^{(l)} := b^{(l)} - \alpha[\frac{1}{m}\nabla_{b^{(l)}} J(W,b)] \end{cases} \tag{16}$$

其中，α 为学习率，一般取 0 ~ 1 的数值。按照上述步骤迭代，直至损失函数小于某个很小的值或超出最大迭代次数，混合自动编码器训练结束。利用混合自动编码器对输入数据进行第一阶段的特征提取，把特征传入第二阶段的分类器训练。混合自动编码器只对原自动编码器的输出层做出改变，而输出层在第二阶段的分类训练中并未保留，所以整个分类模型的结构仍然不变，参见图 2。

3 实验结果与分析

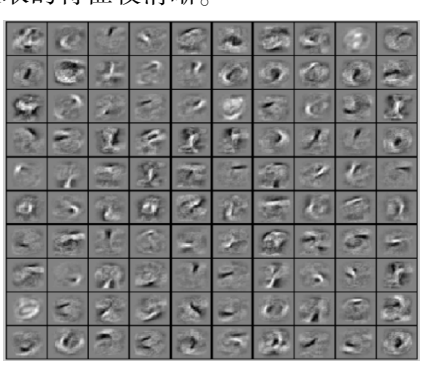
实验平台基于 Matlab 8.2 软件，Windows 8 操作系统，4G 内存的环境运行。限于当前实验平台的限制，没有做相关的深度模型试验。实验数据选取 3 个通用的分类数据集来测试混合自动编码器的分类性能，作为对比将采用改进前后的模型在相同的数据集上进行实验。

3.1 MNIST

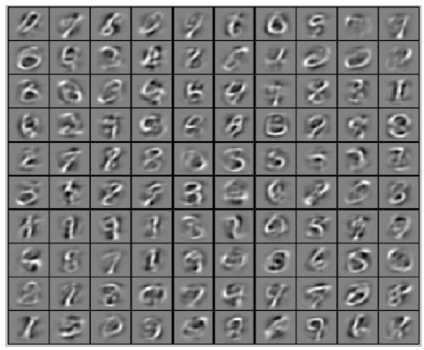
MNIST[22] 手写数字数据集由 0 ~ 9 共 10 种手写数字组成，包含 60000 个训练样本和 10000 个测试样本，每个样本为 28×28 大小的灰度图像，部分图像如图 4 所示。

图 4 MNIST 手写数字集

将隐含层对 MNIST 提取的特征进行可视化，自动编码器(AE)可视化结果如图 5(a)，混合自动编码器(HAE)可视化结果如图 5(b)。前后对比，混合自动编码器提取的特征较清晰。



(a) AE 对 MNIST 的特征可视化



(b) HAE 对 MNIST 的特征可视化

图 5 迭代 400 次，AE 与 HAE 对 MNIST 的特征可视化

表 1 改进前后模型对 MNIST 数据集的实验对比

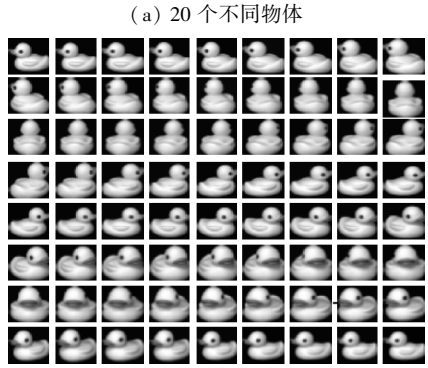
模型	准确率/%	参数及配置
AE	96.640000	神经元个数:784,200,784; 迭代次数:400 次
HAE	98.010000	神经元个数:784,200,784; 迭代次数:400 次

分类结果及对比如表 1 所示，相同参数和配置的情况下，混合自动编码器在 MNIST 数据集上准确率提高了约 1.5%。

3.2 COIL-20

COIL-20[23] 数据集由 20 个不同物体经过 360° 全方位选取不同角度的图片，每个物体包含 72 个不同角度、无背景的图像，共 1440 个样本，每个样本大小为 32×32。20 个物体图像如图 6(a)所示，其中一个物体不同角度图像如图 6(b)所示。数据集划分为 1200 个训练样本和 240 个测试样本。

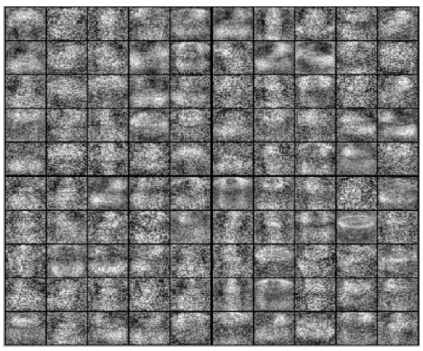
(a) 20 个不同物体



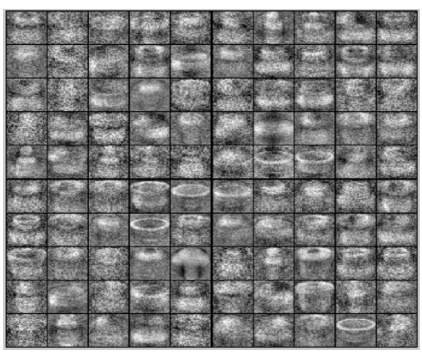
(b) 同一物体不同角度

图 6 COIL-20 数据集

将隐含层对 COIL-20 的特征可视化,自动编码器结果如图 7(a),混合自动编码器结果如图 7(b),混合自动编码器学习到的特征更明晰。



(a) AE 对 COIL-20 的特征可视化



(b) HAE 对 COIL-20 的特征可视化

图 7　迭代 100 次,AE 与 HAE 对 COIL-20 的特征可视化

分类结果及对比如表 2 所示,迭代 100 次,改进后的模型准确率已接近 100%,效果明显。

表 2　改进前后模型对 COIL-20 的实验对比

模型	准确率/%	参数及配置
AE	95.833333	神经元个数:1024,200,1024; 迭代次数:100 次
HAE	99.583333	神经元个数:1024,200,1024; 迭代次数:100 次

3.3　letter-recognition

letter-recognition[24] 数据集由 26 个大写字母的黑白图片组成,如图 8[25] 所示。每个字符图像基于 20 种不同字体,并且每个字母被随机扭曲成不同的外形,共 20000 个样本,每个字符图像由 16 个数值属性(统计值和边缘信息)表达。通常把数据集分为 16000 个训练样本和 4000 个测试样本。

图 8　letter recognition 部分字符图像

由于样本数据不是以像素形式存储,可视化效果模糊,这里不做特征显示。实验结果及对比如表 3 所示,相同参数和配置情况下迭代 400 次,错误率降低了近17 %,效果显著。

表 3　改进前后模型对 letter recognition 的实验对比

模型	准确率/%	参数及配置
AE	76.3000	神经元个数:16,60,26; 迭代次数:400 次
HAE	93.6000	神经元个数:16,60,26; 迭代次数:400 次

以上 3 组数据集的对比试验均表明,对自动编码器加入分类训练后,分类结果都达到了较高的精度。

4　结束语

自动编码器属于无监督学习模型,学习到的特征虽然能有效地重构原始信息,但学习的特征与类别无关,不利于后续的分类操作。为增强自动编码器判别能力,引入 Softmax 分类器,进行有监督训练,使改进后的混合自动编码器在保证重构误差小的同时,提取出更有利于分类的特征,进而提高分类准确率。实验结果表明,混合自动编码器应用于黑白或灰度图像的分类中,分类性能显著提高。进一步的工作将会基于该模型的深度结构进行实验,研究其深度模型的特征表达能力。该模型的缺点是要求样本必须带标注才能进行有监督训练,增加了样本采集的难度。

参考文献:

[1]　高振涛. 判别自动编码机在分类问题中的应用[M]. 成都:四川大学出版社,2015.

[2]　Cortes C, Vapnik V. Support-Vector Networks[J]. Machine Learning, 1995, 20(3): 273-297.

[3]　Safavian S R, Landgrebe D. A survey of decision treeclassifier methodology[J], 1990.

[4]　Liaw A, Wiener M. Classification and regression by randomForest[J]. R news, 2002, 2(3): 18-22.

[5]　Elkan C. Boosting and naive Bayesian learning [R]. Technical Report CS97-557, University of California, San Diego, 1997.

[6]　Arel L, Rose D C, Karnowski T P. Deep machine learning a new frontier in artificial intelligence research[J]. Computational Intelligence Magazine, 2010, 5(4): 13-18.

[7]　Hinton G E, Osinder S, The Y W. A fast learing

algorithm for deep belief nets[J]. Neural Computation, 2006,18(7):1527-1554.

[8] Bengio Y. Learning deep architectures for AI[J]. Foundations and trends in Machine Learning, 2009, 2(1): 1-127.

[9] 倪嘉成,许悦雷,马时平,等.结合侧抑制机制的自动编码器训练新算法[J].计算机应用与软件,2015,(9).

[10] 王雅思.深度学习中的自编码器的表达能力研究[D].哈尔滨:哈尔滨工业大学,2014.

[11] 王勇,赵俭辉,章登义,等.基于稀疏自编码深度神经网络的林火图像分类[J].计算机工程与应用,2014,50(24).

[12] Bengio Y, Lamblin P, Popovici D, et al. Greedy layerwise training of deep networks[C]. Proc. of the 20th Annual Conference on Neural Information Procesing System. 2006:153-160.

[13] Vincent p, Larochelle H, Bengio Y. Extracting and composing robust features with denoising auto-encoder[C]. Proc. of the 25th International Conference on Machine Learning. 2008:1096-1103.

[14] Salalh R, Vincent P, Muller X, et al. Contractive auto-encoders: Explicit invariance during feature extractiong[C]. Proc. of the 28th International Conference on Machine Learning. 2011: 833-840.

[15] Masci J, Meier U, Ciresan D. Stacked convolutional auto-encoders for hierachical feature extraction [C]. Proc. of the 21h International Conference on Artifical Neural Networks. 2011,6791:52-59.

[16] Arnold L, Paugam Moidy H, Sebag M. Unsupervised layerwise model selection in deep neural networks[J]. Journal of Machine Learning Research,2010,6(3) : 610-634.

[17] Erahan D, Bengio Y, Courville A, et al. Why does unsupervised pre-traininghelp deep learning [J]. Journal of Machine Learning Research, 2010,1(5):625-660.

[18] Eehan D, Manzagol P, Bengio Y, et al. The difficulty of training deep architectures and the effect of unsupervised pretraining[C]./Proc of the 12th International Conference on Artificial Intelligence and Statistics. 2009: 153-160.

[19] 胡侯立,魏维,谢青松。深层自动编码机的文本分类算法改进[J].计算机应用研究,2015,32(4).

[20] Ng A, Ngiam J, Foo C Y, et al. UFLDL tutorial. 2012.

[21] Deep Learning Tutorial[EB/OL]. http://deeplearning.net/tutorial/deeplearning.pdf.

[22] Lecun Y, Cortes C. The MNIST database of handwritten digits,1998.

[23] S A Nene, S K Nayar, H Murase. Technical Report CUCS-005-96, Columbia Object Image Library (COIL-20), February 1996.

[24] David J. Slate, Odesta Corporation; Letter Recognition Dataset of the Machine Learning Repository, 1890.

[25] Peter W Frey, David J Slate. Letter recognition using holland-style adaptive calssifiers[J]. Machine Learning,1991,(6):161-182.