

# 浙江工业大学

## 本科毕业设计开题报告

(2015届)



论文题目： 稀疏惩罚因子对无监督学习算法  
的自动编码器的研究分析

作者姓名： 姜楠

指导教师： 龙胜春 副教授

学科(专业)： 计算机科学与技术+自动化1101

所在学院： 计算机科学与技术学院

提交日期： 2015年2月

# 稀疏惩罚因子对无监督学习算法的自动编码器的研究分析

## 一、 选题的背景与意义

### 1.1 研究开发的目的

监督学习在许多领域有很成功的应用，如车牌识别、人脸识别、语音识别、自动驾驶等。尽管如此，监督学习还是有很大的限制。因为监督学习需要手工的选择特征来训练算法。特征的好坏直接影响到学习算法的性能。虽然很多权威学者针对特定的应用提出了很多有效的特征，但对于其他应用，这些特征并不适用。BP神经网络是有效的监督学习算法之一，它具有较强的非线性拟合能力和容错能力，能自动学习数据中的“合理规则”，并将其植入网络权值中，自适应效果好。但是，它受到人工特征提取的限制，步骤繁琐，运算量巨大，代价函数复杂且不易优化，因而收敛速度缓慢，并且该算法易陷入局部极值，泛化能力差，不能精确拟合部分高维复杂函数<sup>[1]</sup>。

有别于监督学习，无监督学习在学习训练时并不知道其分类结果是否正确，亦即没有受到监督式增强(告诉它何种学习是正确的)。其特点是仅对此种网络提供输入样例，它会自动从这些样例中找出其潜在类别规则、特征。当学习完毕并经测试后，也可以将之应用到新的测试集上。稀疏编码算法是一种无监督学习方法，它用来寻找一组“超完备”基向量来更高效地表示样本数据。其目的就是找到一组基向量 $\phi_i$ ，使得我们能将输入向量 $\mathbf{X}$ 表示为这些基向量的线性组合： $\mathbf{X} = \sum_{i=1}^k a_i \phi_i$ ，这组基向量能表示完备地输入向量 $\mathbf{x} \in \mathbb{R}^n$  (也就是说， $k > n$ )。好处是它们能更有效地找出隐含在输入数据内部的结构与模式。然而，对于超完备基来说，系数 $a_i$ 不再由输入向量 $\mathbf{x}$ 唯一确定。因此，在稀疏编码算法中，我们另加了一个评判标准“稀疏性”来解决因超完备而导致的退化 (degeneracy) 问题。为了解决上述问题，Hinton等人提出了稀疏自编码 (sparse autoencoder) 学习算法。它可以用来从未标注的数据中自动学习特征，免去了人工提取数据特征的巨大工作量，提高了特征提取的效率，降低了原始输入的维度，得到数据的逆向映射特征，展现了从少数有标签样本和大量无标签样本数据中学习输入数据本质特征的强大能力，并将学习到的特征分层表示，为构建深度结构奠定了基础。

## 1.2 国内外研究发展现状

1. 1959年, David Hubel 和Torsten Wiesel<sup>[2]</sup>通过对猫的视觉条纹皮层简单细胞感受野的研究得出一个结论: 主视皮层V1区神经元的感受野能对视觉感知信息产生一种“稀疏表示”, 基于这一知识, 1961年, H.B.Barlow<sup>[3]</sup>提出了“利用感知数据的冗余”进行编码的理论。
2. 1969年, D.J.Willshaw和O.P.Buneman<sup>[4]</sup>等人提出了基于Hebbian学习的局部学习规则的稀疏表示模型, 这种稀疏表示可以使模型之间有更少的冲突, 从而使记忆能力最大化, Willshaw 模型的提出表明了稀疏表示非常有利于学习神经网络的联想。
3. 1972年, Barlow<sup>[5]</sup>推论出在稀疏性 (Sparsity) 和自然环境的统计特性之间必然存在某种联系。随后, 有许多计算方法被提出来论证这个推论, 这些方法都成功表明了稀疏表示可以体现出在大脑中出现的自然环境的统计特征。
4. 1986年, Rumelhart提出了自动编码器的概念, 并将其用于高维复杂数据处理, 促进了神经网络的发展<sup>[6]</sup>。
5. 1987年, Field<sup>[7]</sup>提出主视皮层V1区简单细胞的感受野非常适于学习视网膜成像的图像结构, 因为它们可以产生图像的稀疏表示, 基于这个结论, 1988年, Michison明确提出了神经稀疏编码的概念<sup>[8]</sup>, 然后由牛津大学的E.T.Roll 等人正式引用。
6. 1989年, D.J.Field<sup>[9]</sup>提出了稀疏分布编码方法。这种编码方法使响应于任意特殊信息的神经细胞数目被减少, 信号的稀疏编码寻在于细胞响应分布的四阶矩即峭度中。
7. 1996年, B.A.Olshausen和D.J.Field<sup>[10]</sup>在Nature发表了一篇题为Emergence of simple cell receptive field properties by learning a sparse code for natural images 的重要论文, 指出自然图像经过稀疏编码后得到的基函数类似于V1 区简单细胞的感受野的反应特性。
8. 1997年, B.A.Olshausen和D.J.Field<sup>[11]</sup>又提出了一种超完备的稀疏编码算法。
9. 1997年, Bell和Sejnowski<sup>[12]</sup>把ICA用于自然图像分析, 并且得出一个重要的结论: ICA实际上就是一种特殊的稀疏编码。
10. 2006年, Hinton对原型自动编码器的结果进行改进, 进而产生了深度自动编码器, 先用无监督逐层贪心训练的算法完成对隐含层的预训练, 然后用BP算法对整个神经网络进行系统性能参数调整, 显著降低了神经网络的性能指数, 有效改善了BP算法易陷入局部最小的不良情况<sup>[13]</sup>。

11. 2007年, Benjio提出了稀疏自动编码器的概念, 进一步深化了自动编码器的研究<sup>[14]</sup>。
12. 2008年, Vincent提出了降噪自动编码器, 在输入数据中添加噪声向量, 防止出现过拟合现象, 并取得了良好的效果<sup>[15]</sup>。
13. 2009年, Benjio总结了已有的深度结构, 阐述了利用堆叠自动编码器构建深度学习神经网络的一般方法<sup>[16]</sup>。
14. 2010年, salah对升维和降维的过程加以限制, 提出了收缩自动编码器<sup>[17]</sup>。
15. 2011年, Jonathan提出了卷积自动编码器, 用于构建卷积神经网络<sup>[18]</sup>。
16. 2012年, Taylor对深度自动编码器与无监督特征学习之间的联系进行了深入的探讨, 详细介绍了如何利用自动编码器构建不同类型的深度结构<sup>[19]</sup>。
17. 2012年, Hinton, Bengio和Vincent等人对比了原型自动编码器、稀疏自动编码器、降噪自动编码器、收缩自动编码器、卷积自动编码器和限制玻尔兹曼机等结构的性能, 为以后的实践和科研提供了参考<sup>[20-24]</sup>。
18. 2013年, Telmo研究了用不同代价函数训练的自动编码器的性能, 为代价函数的优化策略的发展指明了方向<sup>[25]</sup>。

## 二、 研究开发的基本内容、目标

### 2.1 研究目标

基于稀疏理论的自动编码器是目前应用最为广泛的自动编码器, 它能提取高维数据变量的稀疏解释性因子, 保留原始输入的非零特征, 增加表示算法的鲁棒性, 增强数据的线性可分性, 是分类边界变得更加清晰, 并且能在一定程度上空盒子变量的规模, 改变给定输入数据结构, 丰富了原有信息, 提高了信息表述的全面性和准确率。但由于原始数据分布稠密程度不同, 经过信息解锁后的稀疏变量难以控制。本课题的研究目标定位于利用Matlab技术来实现自学习算法(稀疏自动编码器+Softmax回归模型), 着重于讨论并研究稀疏自动编码器中的稀疏惩罚因子(Sparsity Penalty)对该自学习算法的性能表现, 我们主要按照三个指标来评价其性能差异: 提取的特征的稀疏度, 测试集上的重构误差, 测试集上的精确度。通过研讨其在不同数据集上的性能表现, 期望分析出稀疏惩罚因子的实用价值及效用。

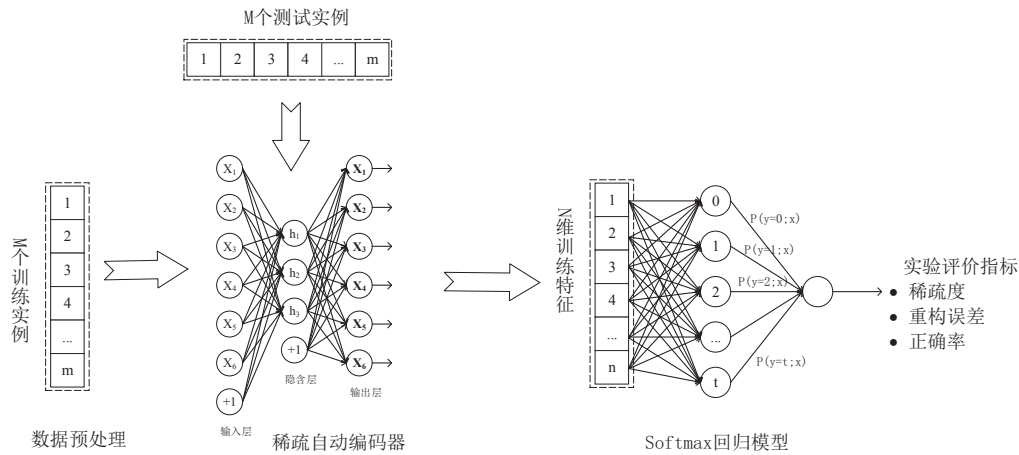


图 2.1 实验研究的基本框架

## 2.2 研究的基本内容

实验框架如图2.1所示，主要由4个部分组成：数据预处理、稀疏自编码器训练、Softmax回归模型训练、统计计算实验指标。下面将详细解释了各个部分的具体内容。

● **数据预处理：**数据预处理中，标准的第一步是数据归一化。特征归一化常用的方法包含如下几种：简单缩放、逐样本均值消减、特征标准化(使数据集中所有特征都具有零均值和单位方差)。

(1) **简单缩放：**在简单缩放中，我们的目的是通过对数据的每一个维度的值进行重新调节（这些维度可能是相互独立的），使得最终的数据向量落在 $[0,1]$ 或 $[-1,1]$ 的区间内（根据数据情况而定）。在处理自然图像时，我们获得的像素值在 $[0,255]$ 区间中，常用的处理是将这些像素值除以255，使它们缩放到 $[0,1]$ 中。

(2) **逐样本均值消减：**如果数据是平稳的（即数据每一个维度的统计都服从相同分布），那么可以考虑在每个样本上减去数据的统计平均值(逐样本计算)。

(3) **特征标准化：**特征标准化指的是（独立地）使得数据的每一个维度具有零均值和单位方差。

● **稀疏惩罚因子的种类：**本实验拟采用机器学习中常用的惩罚因子： $L1$ 范数和 $L2$ 范数，还有统计学中常用的数学模型： $\text{student-t}$ 分布模型，最后是稀疏自编码模型中常用的惩罚因子： $\text{KL}$ 距离（Kullback-Leibler divergence），作为主要的惩罚因子类别，并讨论分析其在实验中的性能表现。

● **稀疏自编码器训练：**自动编码器是一种利用无监督预训练的非线性网络结构，从无类表数据中提取高维复杂输入数据的分层特征，并得到原始数据的分布式特征表示的深度神经网络结构<sup>[26]</sup>。它由编码器、解码器组成。编码器是输入 $x$ 到隐含层特征 $h$ 的映射，

解码器是隐含层将数据映射回重构的输入层。训练自动编码器的过程是在训练样本集 $D$ 上寻找 $\theta = \{W, b_y, b_h\}$ 的最小化重构误，一般可用平方误差函数或交叉熵损失函数。基于稀疏理论的自动编码器对原型自动编码器的隐含层添加了约束条件并增加了隐含层数量。我们拟采用神经网络中的梯度下降算法来训练稀疏自动编码器。

● **Softmax回归模型训练：**Softmax回归模型是logistic回归模型在多分类问题上的推广，在多分类问题中，类标签 $y$ 可以取两个以上的值。Softmax回归模型对于诸如MNIST手写数字分类等问题是很有用的，该问题的目的是辨识10个不同的单个数字，Softmax回归是有监督的。我们也来用梯度下降算法来训练Softmax回归模型 $J(\theta; \lambda)$ ，但在训练该模型的同时，我们还需要通过K层交叉验证手段配置合适的超参数 $\lambda$ ，以获得最优的模型参数配置。

● **统计计算实验指标：**当我们利用训练数据集完成了稀疏自编码器和Softmax回归模型的训练后，我们将采用测试数据集，按顺序输入稀疏自编码器和Softmax回归模型，与此同时我们需要统计：稀疏自编码器中获得的特征稀疏度、稀疏自编码器在测试集上的平均重构误差、Softmax模型的识别准确率。我们拟采用五个图像数据集，分别输入模型，再收集不同数据集上的数据后，分析其性能表现。

## 2.3 需要解决的关键技术问题

掌握并理解深度学习理论中自动编码器、Softmax回归模型的原理及理论。在理解并实践深度学习理论之前，需要对机器学习领域的算法有所掌握，例如：线性回归模型，梯度下降迭代算法。基础知识熟练程度与否关系着日后实验中是否能合理解决遇到的学术问题。

● 配置Linux上Matlab开发环境并掌握使用Matlab Script和Linux常用指令集。

(1) Matlab是由美国MathWorks公司开发的商业数学软件，主要用于算法开发、数值计算和数据分析，在工程控制、信号处理和金融建模等领域中均有应用。Matlab软件对内存需求极大，其内部的矩阵运算和数据存储严重耗费内存，同时实验研究中采用的数据集大小均在10,000 ~ 100,000之间，故需要配置并采用X64架构的Matlab开发软件。另一方面，个人笔记本或者PC已无法满足数值计算所需的硬件基础，故本次实验研究拟采用实验室的Centos服务器作为硬件基础开发载体。其32核IntelXeon CPU和64GB大容量内存足以满足实验运行要求。考虑到对Linux开源操作系统和对Linux下配置Matlab开发环境不熟悉，需要进一步实践并努力掌握其常用terminal终端控制指令和软件使用方法。

(2) Matlab中虽然包含了神经网络工具箱，但尚未集成深度学习的工具箱。由于深度学习的概念提出较新，虽然Github上存在许多版本的开发扩展库，但其质量参差不齐。

在学者尚未给出权威参考版本之前，参考引用时需要加以甄别。另一方面，也对Matlab Script和丰富的内置函数的使用与调试提出了较高要求。仔细研读实验代码框架，并认真检查、调试实验核心代码是本次实验研究必须达到的要求。

- 搭建稀疏自动编码器模型，并挑选合适的数据集并对其做数据预处理。数据的尺寸、数据集样本的数量是无监督学习中重要的参数，若选择的数据尺寸过大，将导致预算过于缓慢，函数收敛速度过慢，需要较长的训练时间。若选择数据集的样本数量过小，将使得无监督学习的结果产生较大的泛化误差，实验数据数量较少，从而学习到的特则会难过不具有一般性，在测试集上正确率降低。若实验数据集样本过多，数据尺寸较大，则实验硬件环境将无法承受其运算要求，内存将长期处于高负荷状态，CPU利用率较低，耗费的训练时间相对更长。

### 三、 研究开发的方法、技术路线和步骤

(1) 系统平台：centos 6，Intel Xeon 24核，64GB内存。CentOS<sup>1</sup>（Community Enterprise Operating System）是Linux发布版之一，它是来自于Red Hat Enterprise Linux依照开放源代码规定发布的源代码所编译而成。由于出自同样的源代码，因此有些要求高度稳定性的服务器以CentOS替代商业版的Red Hat Enterprise Linux使用。两者的不同，在于CentOS并不包含封闭源代码软件。CentOS 对上游代码的主要修改是为了移除不能自由使用的商标。

(2) 编程语言：Matlab Script，Java JDK支持。在配置Matlab之前，需要配置Sun Java JDK<sup>2</sup>开发环境。由于Linux/Unix操作系统默认配置open JDK作为替代Sun Java JDK的java开发包，这在通用依赖java的软件中尚未构成问题。然而例如Matlab，Eclipse，Hadoop等软件必须依赖Sun Java JDK开发包和Sun Java JRE开发环境，故需要配置linux下Sun Java开发环境，以便后续步骤顺利进行。采用Matlab<sup>3</sup>作为主要开发工具，主要基于以下几点考虑：

(a) 编程环境：MATLAB由一系列工具组成。这些工具方便用户使用MATLAB的函数和文件，其中许多工具采用的是图形用户界面。包括MATLAB桌面和命令窗口、历史命令窗口、编辑器和调试器、路径搜索和用于用户浏览帮助、工作空间、文件的浏览器。随着MATLAB的商业化以及软件本身的不断升级，MATLAB的用户界面也越来越精致，更加接近Windows的标准界面，人机交互性更强，操作更简单。而且新版本的MATLAB提供了完整的联机查询、帮助系统，极大的方便了用户的使用。简单的编程环境提供了比较完备的调试系统，程序不必经过编译就可以直接运行，而且能够及时地报告出现的错误及进行

<sup>1</sup><http://www.centos.org/>

<sup>2</sup><http://www.oracle.com/index.html>

<sup>3</sup><http://cn.mathworks.com/>

出错原因分析。

**(b) 简单易用：**Matlab是一个高级的矩阵/阵列语言，它包含控制语句、函数、数据结构、输入和输出和面向对象编程特点。用户可以在命令窗口中将输入语句与执行命令同步，也可以先编写好一个较大的复杂的应用程序（M文件）后再一起运行。新版本的MATLAB语言是基于最为流行的C++语言基础上的，因此语法特征与C++语言极为相似，而且更加简单，更加符合科技人员对数学表达式的书写格式。使之更利于非计算机专业的科技人员使用。而且这种语言可移植性好、可拓展性极强，这也是MATLAB能够深入到科学研究及工程计算各个领域的重要原因。

**(c) 强处理能力：**MATLAB是一个包含大量计算算法的集合。其拥有600多个工程中要用到的数学运算函数，可以方便的实现用户所需的各种计算功能。函数中所使用的算法都是科研和工程计算中的最新研究成果，而前经过了各种优化和容错处理。在通常情况下，可以用它来代替底层编程语言，如C和C++。在计算要求相同的情况下，使用MATLAB的编程工作量会大大减少。MATLAB的这些函数集包括从最简单最基本的函数到诸如矩阵，特征向量、快速傅立叶变换的复杂函数。函数所能解决的问题其大致包括矩阵运算和线性方程组的求解、微分方程及偏微分方程的组的求解、符号运算、傅立叶变换和数据的统计分析、工程中的优化问题、稀疏矩阵运算、复数的各种运算、三角函数和其他初等数学运算、多维数组操作以及建模动态仿真等。

**(d) 图形处理：**MATLAB自产生之日起就具有方便的数据可视化功能，以将向量和矩阵用图形表现出来，并且可以对图形进行标注和打印。高层次的作图包括二维和三维的可视化、图象处理、动画和表达式作图。可用于科学计算和工程绘图。新版本的MATLAB对整个图形处理功能作了很大的改进和完善，使它不仅在一般数据可视化软件都具有的功能（例如二维曲线和三维曲面的绘制和处理等）方面更加完善，而且对于一些其他软件所没有的功能（例如图形的光照处理、色度处理以及四维数据的表现等），MATLAB同样表现了出色的处理能力。同时对一些特殊的可视化要求，例如图形对话等，MATLAB也有相应的功能函数，保证了用户不同层次的要求。另外新版本的MATLAB还着重在图形用户界面（GUI）的制作上作了很大的改善，对这方面有特殊要求的用户也可以得到满足。

MATLAB对许多专门的领域都开发了功能强大的模块集和工具箱。一般来说，它们都是由特定领域的专家开发的，用户可以直接使用工具箱学习、应用和评估不同的方法而不需要自己编写代码。领域，诸如数据采集、数据库接口、概率统计、样条拟合、优化算法、偏微分方程求解、神经网络、小波分析、信号处理、图像处理、系统辨识、控制系统设计、LMI控制、鲁棒控制、模型预测、模糊逻辑、金融分析、地图工具、非线性控制设计、实时快速原型及半物理仿真、嵌入式系统开发、定点仿真、DSP与通讯、电力系统仿



真等，都在工具箱（Toolbox）家族中有了自己的一席之地。

**(a) 程序接口：**新版本的MATLAB可以利用MATLAB编译器和C/C++数学库和图形库，将自己的MATLAB程序自动转换为独立于MATLAB运行的C和C++代码。允许用户编写可以和MATLAB进行交互的C或C++语言程序。另外，MATLAB网页服务程序还容许在Web应用中使用自己的MATLAB数学和图形程序。MATLAB的一个重要特色就是具有一套程序扩展系统和一组称之为工具箱的特殊应用子程序。工具箱是MATLAB函数的子程序库，每一个工具箱都是为某一类学科专业和应用而定制的，主要包括信号处理、控制系统、神经网络、模糊逻辑、小波分析和系统仿真等方面的应用。

**(b) 应用软件开发：**在开发环境中，使用户更方便地控制多个文件和图形窗口；在编程方面支持了函数嵌套，有条件中断等；在图形化方面，有了更强大的图形标注和处理功能，包括对性对起连接注释等；在输入输出方面，可以直接向Excel和HDF5进行连接。

#### 四、 研究工作总体安排与时间进度

表 4.1 研究工作总体安排与时间进度表

任务序号	起止时间	任务阶段要点
1	2015.1.1-2015.2.1	了解课题相关内容、查找中、英资料
2	2015.2.2-2015.2.28	查阅文献资料，完成文献综述、 开题报告和外文翻译
3	2015.3.1-2015.3.15	学习机器学习与深度学习相关理论
4	2015.3.16-2015.3.31	搭建系统开发框架，配置开发环境
5	2015.4.1-2015.4.15	进行功能模块测试
6	2015.4.16-2015.4.30	完成系统框架，配置实验参数，开展实验
7	2015.5.1-2015.5.25	检查数据错误，重做错误结果， 统计整理数据，分析实验结果
8	2015.5.26-2015.6.4	整理资料、完成毕业论文
9	2015.6.5-2015.6.10	上交毕业论文、准备毕业答辩

## 参考文献

- [1] Itamar Arel, Derek C Rose, Thomas P Karnowski. Deep machine learning-a new frontier in artificial intelligence research [research frontier][J]. Computational Intelligence Magazine, IEEE, 2010. 5(4):13–18.
- [2] David H Hubel, Torsten N Wiesel. Receptive fields of single neurones in the cat's striate cortex[J]. The Journal of physiology, 1959. 148(3):574–591.
- [3] Horace B Barlow. The coding of sensory messages[J]. Current problems in animal behaviour, 1961:331–360.
- [4] David J Willshaw, O Peter Buneman, Hugh Christopher Longuet-Higgins. Non-holographic associative memory.[J]. Nature, 1969.
- [5] Horace B Barlow. Single units and sensation: a neuron doctrine for perceptual psychology?[J]. Perception, 2009. (38):795–8.
- [6] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams. Learning internal representations by error propagation[R]. :DTIC Document, 1985.
- [7] David J Field. Relations between the statistics of natural images and the response properties of cortical cells[J]. JOSA A, 1987. 4(12):2379–2394.
- [8] G Mitchinson. The organization of sequential memory: Sparse representations and the targeting problem[C]//Proceedings of the Bad Homburg meeting on Brain Theory. volume 1619. 1986.
- [9] David J Field. What the statistics of natural images tell us about visual coding[C]//OE/LASE'89, 15-20 Jan., Los Angeles. CA. International Society for Optics and Photonics, 1989:269–276.
- [10] Bruno A Olshausen, et al. Emergence of simple-cell receptive field properties by learning a sparse code for natural images[J]. Nature, 1996. 381(6583):607–609.
- [11] Bruno A Olshausen, David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1?[J]. Vision research, 1997. 37(23):3311–3325.
- [12] Anthony J Bell, Terrence J Sejnowski. The “independent components” of natural scenes are edge filters[J]. Vision research, 1997. 37(23):3327–3338.
- [13] Geoffrey Hinton, Simon Osindero, Yee-Whye Teh. A fast learning algorithm for deep belief nets[J]. Neural computation, 2006. 18(7):1527–1554.
- [14] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al. Greedy layer-wise training of deep networks[J]. Advances in neural information processing systems, 2007. 19:153.
- [15] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders[C]//Proceedings of the 25th international conference on Machine

- hr/>
- learning. ACM, 2008:1096–1103.
- [16] Yoshua Bengio. Learning deep architectures for ai[J]. Foundations and trends® in Machine Learning, 2009. 2(1):1–127.
- [17] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction[C]//Proceedings of the 28th International Conference on Machine Learning (ICML-11). 2011:833–840.
- [18] Jonathan Masci, Ueli Meier, Dan Cireşan, Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. //Artificial Neural Networks and Machine Learning–ICANN 2011. Springer, 2011. 52–59.
- [19] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures.[J]. ICML Unsupervised and Transfer Learning, 2012. 27:37–50.
- [20] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion[J]. The Journal of Machine Learning Research, 2010. 11:3371–3408.
- [21] Ben Mitchell, John Sheppard. Deep structure learning: Beyond connectionist approaches[C]//Machine Learning and Applications (ICMLA), 2012 11th International Conference on. volume 1. IEEE, 2012:162–167.
- [22] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, Samy Bengio. Why does unsupervised pre-training help deep learning?[J]. The Journal of Machine Learning Research, 2010. 11:625–660.
- [23] Li Deng, Michael L Seltzer, Dong Yu, Alex Acero, Abdel-Rahman Mohamed, Geoffrey E Hinton. Binary coding of speech spectrograms using a deep auto-encoder.[C]//Interspeech. Citeseer, 2010:1692–1695.
- [24] Honglak Lee, Chaitanya Ekanadham, Andrew Y Ng. Sparse deep belief net model for visual area v2[C]//Advances in neural information processing systems. 2008:873–880.
- [25] Telmo Amaral, Luís M Silva, Luís A Alexandre, Chetak Kandaswamy, Jorge M Santos, Joaquim Marques de Sá. Using different cost functions to train stacked auto-encoders[C]//Artificial Intelligence (MICAI), 2013 12th Mexican International Conference on. IEEE, 2013:114–120.
- [26] Yoshua Bengio, Olivier Delalleau. On the expressive power of deep architectures[C]//Algorithmic Learning Theory. Springer, 2011:18–36.