

# 浙江工业大学

## 学士学位论文

(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 6月

## 摘要

这是摘要

关键词： 关键词1， 关键词2

## Abstract

The quick brown fox jump over the lazy dog.

**Keywords:** keywords1,keywords2,

## 插图

2.1 实验研究的基本框架.....	9
--------------------	---

## 表格

4.1 研究工作总体安排与时间进度表.....	13
-------------------------	----

## 目录

摘要 .....	1
Abstract.....	2
插图 .....	3
表格 .....	4
目录 .....	
第一章 选题的背景与意义.....	6
1.1 研究开发的目的 .....	6
1.2 国内外研究发展现状 .....	7
第二章 研究开发的基本内容、目标.....	9
2.1 研究目标 .....	9
2.2 研究的基本内容 .....	9
2.3 需要解决的关键技术问题 .....	11
第三章 研究开发的方法、技术路线和步骤.....	12
第四章 研究工作总体安排与时间进度.....	13
参考文献 .....	14
致 谢 .....	16

# 第一章 选题的背景与意义

## 1.1 研究开发的目的

监督学习（supervised learning）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。提取的特征质量直接影响到机器学习算法的性能。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，但这些算法技巧不具有推广的意义。BP神经网络模型由输入层、隐藏层、输出层构成，是监督学习算法中的一个广泛使用的模型。通过求解误差的偏导，并利用反向传导公式迭代求解出系统的极值，从而学习出数据集中的“合理规则”。特别地，他能突破传统多项式公式有限的拟合数据的能力，能学习到更加复杂的非线性函数规律和即使数据中包含一些误差，他也能收敛致最优解，拥有较强的自适应性。然而，由于BP神经网络的代价函数不是严格意义上的凸函数，一旦算法在迭代优化过程中陷入局部极值，就有可能无法求得系统的最优解。若要拟合高维复杂的函数，采用较复杂的神经网络模型就会导致函数运算量巨大、收敛过慢，若采用较小的神经网络模型，则只能学习到简单的函数方程，可能会忽略了重要的信息特征<sup>[1]</sup>。

然而在实际应用中，我们无法预先知道样本的标签，只能从原始的无标签的样本集开始，进行分类器的设计，即通常所说的无监督学习方法（unsupervised learning）。无监督学习方法可以分类两类，一类为基于概率密度函数估计的方法，它试图找出数据集在特征空间的分布参数，再用特征空间分布进行分类处理，例如稀疏编码算法。另一类称为基于样本间的相似性度量的聚类方法，它通过聚类方法将样本划分成不同类别。稀疏编码算法是第一类无监督算法的重要实践，它将原始信号表示为字典元素的一个线性组合： $x = D * a$ ，我们用 $x$ 表示原始信号， $D$ 为我们得到的字典（dictionary），通过字典 $D$ ，原始信号 $x$ 能被完备地表示为 $a$ 的线性组合。当我们将稀疏性概念引入后，即我们希望 $a$ 是稀疏的，只有较少的非零项，这样做最突出的优点是运算速度能有效提升，因为Matlab只对非零元素进行操作。另一方面，为了解决字典 $D$ 的退化（degeneracy）问题，Hinton等人提出了稀疏自编码（Sparse Auto-encoder）学习算法。稀疏自编码算法是一种自动提取样本（如图像）特征的方法，它把原始数据用隐藏层的特征空间向量表示，再把隐藏层特征解码成重构的输出层。这样，隐藏层中的特征信息就是输入层的一个压缩表示，而且有效减少了冗余信息。并且，压缩表示的特征很适合进一步用作分类器的输入集。该模型可

以用来标注无标签的数据集，免去了繁复的人工打标签的操作，同时也能有效降低输入数据的维度，防止出现“维度灾难”。

## 1.2 国内外研究发展现状

1. 1959年，David Hubel等人<sup>[2]</sup>在神经实验中，研究猫的视觉条纹皮层简单细胞，得出一个惊人的结论：主视皮层V1区神经元能将视觉信息转化为“稀疏表示”的特征，自此提出机器学习领域重要的“稀疏表示”的概念。
2. 1972年，Barlow<sup>[3]</sup>推导演绎出，自然环境的统计模型和稀疏性（Sparsity）必然存在某种联系，即稀疏表示可以体现出在大脑中的统计特征。
3. 1986年，Rumelhart提出了自动编码器（Auto-encoder）的概念模型，并将此模型用于处理高维复杂数据，展示其模型的性能和优势<sup>[4]</sup>。
4. 1987年，Field等人<sup>[5]</sup>提出，主视皮层V1区细胞在学习视网膜的图像结构后能产生图像的稀疏表示。1989年，D.J.Field等人<sup>[6]</sup>提出了稀疏编码方法，它能使只有少数神经细胞响应任意的信号输入。
5. 1997年，B.A.Olshausen和D.J.Field<sup>[7]</sup>按照集合的完备性的理论，提出了“超完备”的稀疏编码算法。
6. 1997年，Bell和Sejnowski<sup>[8]</sup>利用独立分量分析算法（Independent Component Analysis）分析自然图像，发现独立分量算法也能产生稀疏编码的特征。
7. 2006年，Hinton和他的学生Ruslan Salakhutdinov<sup>[9]</sup>修正了原型自动编码器的模型结构，并称其为深度自动编码器。先用无监督学习算法，逐层贪心训练单层的自动编码器，最后针对整个深度自动编码器，用反向传播算法进行系统的参数调优，这样的训练手段显著降低了神经网络的训练时间，同时也能有效防止传统反向传播方法极易陷入局部极值的情况。
8. 2007年，Benjio分层贪心训练受限玻尔兹曼机，再集中调优深度信念网络，取得了历史最佳的分类精度<sup>[10]</sup>。
9. 2008年，Vincent改进了自动编码器模型，在输入层添加高斯白噪声，获得了降噪自动编码器，它能有效降低系统的过拟合现象，具备更好的鲁棒性和良好的分类精度<sup>[11]</sup>。



- 
10. 2009年, Benjio等人在文章“Learning Deep Architectures for AI”中总结了RBMs, SAE, CNN等已有的深度结构, 并分析了在不同情况下的系统特性<sup>[12]</sup>。
  11. 2010年, salah等人在文章“Contractive Auto-Encoders: Explicit Invariance During Feature Extraction”中, 通过对系统的代价函数中添加Frobenius范数的惩罚因子, 对自动编码器的解码和编码过程加以限制, 提出了收缩自动编码器, 获得了类似于降噪自动编码器、约束自动编码器的实验效果<sup>[13]</sup>。
  12. 2011年, Lugano和Switzerland在文章“Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction”中, 将卷积神经网络架构引入自动编码器模型, 同时消去了不必要的约束惩罚项, 利用其单层卷积自动编码器构建的深层卷积神经网络实现了在权威数据集上的突破性成果<sup>[14]</sup>。
  13. 2012年, Pierre Baldi等人从数学角度, 分析线性自动编码器并推导出非线性自动编码器。同时, 也分析了自动编码器与无监督学习之间的联系<sup>[15]</sup>。
  14. 2012年, Bengio等人比较了原型自动编码器、稀疏自动编码器、卷积自动编码器、收缩自动编码器、降噪自动编码器和限制玻尔兹曼机等结构的在不同参数配置和实验数据集上的性能表现。<sup>[16-20]</sup>。
  15. 2013年, Telmo Amaral在文章“Using Different Cost Functions to Train Stacked Auto-encoders”中, 比较在训练过程中使用不同的代价函数, 并评价分析这些代价函数对特征学习的影响, 指出了优化代价函数的研究方向<sup>[21]</sup>。

## 第二章 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，他能在低维的特征空间内重建高维数据变量，并获得稀疏表示的特征。获得的能用于后续的分类器，能有效增强数据的线性可分性，提高了分类器的准确率。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。

### 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)。

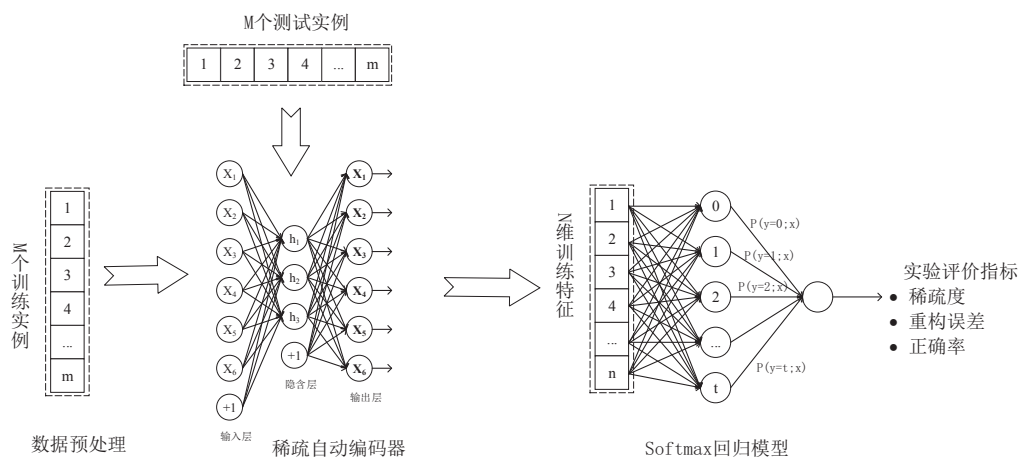


图 2.1 实验研究的基本框架

(1) **简单缩放**: 我们通过简单缩放来重新调节各个维度的值域, 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)。针对图像, 我们将像素在 $[0,255]$ 区间除以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**: 数据项减去其均值, 再除以其方差, 使得每一个维度具有零均值和单位方差, 称为特征标准化。

● **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: student-t分布模型, 最后是稀疏自编码模型中常用的惩罚因子: 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**: 自动编码器是一种无监督学习算法中的一种非线性网络结构, 他能从无标签输入数据中提取出有效的特征信息, 并能作为其他的深度学习网络架构的特征输入<sup>[22]</sup>。自动编码器包含编码过程和解码过程。在编码过程中, 我们将数据输入层 $x$ 映射到隐含层特征 $h$ 。在解码过程中, 我们将隐含层特征 $h$ 映射会重构的输入层 $\hat{x}$ 。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。通过对原型自动编码器添加隐含层的约束, 我们能获得稀疏表示的特征。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**: 传统的logistic回归模型只能适用于二元分类模型, 类标签 $y$ 只能取两个值( $y = \{0, 1\}$ ), 无法解决多元分类问题。在原有基础上, 推导出的Softmax回归模型能解决多元分类问题, 类标签 $y$ 能选择多个值( $y = \{0, 1, \dots, n\}$ )。Softmax回归模型能有效识别手写数字分类问题, 例如MNIST数据集上的数字识别问题, 辨识10个不同的单个数字。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是矩阵实验室的简写（Matrix Laboratory），由MathWorks公司运营，它和Mathematica, SPSS被称为科学计算领域的三大应用，在计算机控制、信号处理等领域中均有重大实践应用。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Intel Xeon 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Bitbucket, Coding.net等知名的开源代码网站上存在许多版本的第三方支持源代码，由于无法评价其质量可靠性。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。

### 第三章 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>是企业级社区操作系统，它是Linux的重要的一个发型系列产品。一直以来，由红帽公司负责维护，开发其更新和修复主要的功能缺陷。当年，红帽公司本着开源的精神毅然将其开发的著名操作系统RHEL中的稳定的软件代码公开，并建立开发者社区来维护和更新其CentOS的系统架构。考虑到RHEL和CentOS拥有同样的源代码，因此有些企业针对部分服务器安装CentOS操作系统以替代昂贵的商业版的RHEL，CentOS的稳定的网络框架秉承RHEL操作系统的特点能满足苛刻的实践应用。RHEL和CentOS操作系统的主要不同点在，于CentOS并不包含RHEL开发的闭源代码库。CentOS对部分源代码进行修改，以便维护商家版权，减少不必要的纷争。

(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：**本次实验不考虑使用集成的Matlab Simulink神经网络工具箱，转而独立使用matlab脚本语言实现算法框架。Matlab软件提供成熟的矩阵运算和丰富的科学计算函数库，极大方便了数学公式的代码实现。与此同时，Matlab针对矩阵运算进行了精密地优化，能并行地计算矩阵间的操作。Matlab的MEX编译器能支持C++，FORTRAN等语言的混合编译，部分对性能要求严格的函数转而利用C语言实现，能在硬件环境下更高效的执行。在分布式计算、大数据等潮流大行其道的今天，Matlab也提供了自身的分布式计算框架：Matlab Distributed Computing Engine（分布式计算引擎）的服务，它能利用计算集群来完成高负载的计算任务。在数据可视化方面，Matlab能利用脚本语言直接操作函数图像的绘制，并对其进行标注、打印。相较于跨平台的开源软甲GNU Octave，matlab能提供更丰富的程序接口和系统计算库，故二者只兼容常见的函数，在许多特殊的地方各异。

---

<sup>1</sup>CentOS: [www.centos.org/](http://www.centos.org/)

<sup>2</sup>Java Development Kits: [www.oracle.com/index.html](http://www.oracle.com/index.html)

<sup>3</sup>Matlab: [cn.mathworks.com/](http://cn.mathworks.com/)

## 第四章 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	了解课题研究点，查阅科技文献
2	2015.2.2-2015.2.28	引用历史文献，提交开题报告、 文献综述、外文翻译
3	2015.3.1-2015.3.15	学习机器学习理论和深度学习概念
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	完善毕业论文，批判吸收文献资料
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. Single units and sensation: a neuron doctrine for perceptual psychology?[J]. Perception, 2009. (38):795–8.
- [4] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams. Learning internal representations by error propagation[R]. :DTIC Document, 1985.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] Anthony J Bell, Terrence J Sejnowski. The “independent components” of natural scenes are edge filters[J]. Vision research, 1997. 37(23):3327–3338.
- [9] Geoffrey Hinton, Simon Osindero, Yee-Whye Teh. A fast learning algorithm for deep belief nets[J]. Neural computation, 2006. 18(7):1527–1554.
- [10] 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.
- [11] 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 learning. ACM, 2008:1096–1103.
- [12] Yoshua Bengio. Learning deep architectures for ai[J]. Foundations and trends® in Machine Learning, 2009. 2(1):1–127.
- [13] 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.
- [14] Jonathan Masci, Ueli Meier, Dan Cireşan, Jürgen Schmidhuber. Stacked convolutional auto-encoders for

- hr/>
- hierarchical feature extraction. //Artificial Neural Networks and Machine Learning–ICANN 2011. Springer, 2011. 52–59.
- [15] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures.[J]. ICML Unsupervised and Transfer Learning, 2012. 27:37–50.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] Yoshua Bengio, Olivier Delalleau. On the expressive power of deep architectures[C]//Algorithmic Learning Theory. Springer, 2011:18–36.



## 致 谢

对给予各类资助、指导、和协助完成研究工作，以及提供各种条件的单位及个人表示感谢。致谢应实事求是，切忌浮夸与庸俗之词。（小四，宋体，段前段后0磅，1.5倍行间距）。