

# 浙江工业大学

## 本科毕业设计外文翻译

(2015届)



论文题目: 无监督学习中单层网络模型的研究分析

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

# 无监督学习中单层网络模型的研究分析

**摘要：** 大量学者都集中在研究从无标签数据中学习特征的算法。事实上，通过使用更复杂的非监督式学习算法和更深层次的模型，学者已经在NORB和CIFAR等数据集上的基准性能测试取得巨大的进步。然而，在本文中，我们尝试证明一些简单的因素，例如模型中隐藏节点的数量，会比学习算法或者模型的深度对系统的表现产生更重要的影响。具体而言，我们将针对CIFAR，NORB，STL数据集采用一些现成的特征学习算法（稀疏自动编码器，稀疏限制玻尔兹曼机，k-means聚类，高斯混合模型）并且均采用单层网络模型。然后我们对系统的指标：感受野大小，隐藏层节点（特征）数目和提取的特征之间的步长（步幅）对系统产生的影响进行了分析，并且进一步分析白化方法对系统的影响。我们的研究表明，大量的隐藏节点和密集的特征提取是实现高性能系统表现必不可少的手段。实际上，当这些因子都调节到自身的极限，我们仅使用单层的特征模型在CIFAR-10和NORB数据集上面都实现了最先进的性能表现。更令人惊讶的是，K-means聚类方法系统不仅运算速度快，而且相对于其他算法模型表现最佳，并且无需调整超出该模型架构的参数，很容易实现。尽管我们的系统架构简洁，我们实现了超越以往所有已发布结果的正确率，CIFAR-10数据集（正确率为：79.6%）和NORB数据集（正确率：97.2%）。

## 一、 简介

最近在机器学习领域中，许多学者正致力于从无标签输入数据中学习获得优良的特征表示，以便于更高层次的任务，如分类。目前的解决方案主要通过非监督式学习算法<sup>[1-3]</sup>贪心地预训练多层次的特征，每次只训练一层，来获得多层次的特征表示。通过如下方法选择每一层的设计参数：需要学习的特征数量，特征被处理的位置，以及如何将系统的输入与输出编码。在本文中，我们研究了在单层网络模型上针对上述不同的选择，采用不同的特征学习方法，并分析其影响效果。我们的研究表明，几个互相不干扰的关键点：白化，大量特征和密集的特征提取，对系统表现有着很重要的影响，是所有算法中的优势所在。即采用简单的算法和单层的特征，而不是更新算法模型架构，也可能通过合理调节这些因子来获得当前最好的系统表现。

很多特征学习算法的主要缺点是其高复杂性和时间复杂性。此外，许多算法需要需要仔细选择多个超参数像学习速率，动量，稀疏惩罚因子，权重衰减因子等等，这些因子需要通过交叉验证手段来挑选，从而急剧增加了系统的运行时间。虽然最近发表的算法一直在改善在NORB<sup>[4]</sup> 和CIFAR-10<sup>[5]</sup> 数据集上面的性能表现，但系统中仍然存在其他因素能

明显影响特征学习算法的系统性能。具体地说，有许多“元参数”能决定系统的网络架构，例如感受野的大小和隐藏节点（特征）的数量。在实践中，这些参数通常由于计算瓶颈限制而预选被确定。例如，考虑到算法的运行时间我们可能会使用最多的特征数量。然而在本文中，我们采用另一种策略：我们使用十分简单的学习算法并且仔细选择网络模型的参数，期望能获得更好的性能表现。如果（在很多情况下）特征数量越多会有更好的表现，那么我们就可以充分这些学习算法的简洁性和运算速度来挑选更多的特征数量。

为此，我们将在第三节介绍一个简单的特征学习框架，它把特征学习算法作为“黑匣子”模块使用。对于该“黑匣子”，我们已经实现了几个现成的非监督学习算法：稀疏自动编码器，系数限制玻尔兹曼机，k-means聚类和高斯混合模型。然后我们分析了在特征学习框架中几种不同的因素对系统的性能的影响，包括：

- 白化，这是深度学习中常用的数据预处理手段；
- 训练的特征数目；
- 提取特征间的步长（步幅）
- 感受野的大小

实验表明，无论采用什么非监督式学习算法，采用白化，大量的特征数量和小的步幅均能产生更好的性能表现。一方面，这些结果有些令人惊异。例如，业界广泛认为大量的特征表示能比采用小尺寸的表示产生更好的性能表现<sup>[6]</sup>，并且对于特征间的小步幅有类似的结论<sup>[7]</sup>。然而，我们工作的主要贡献就是证明这些因素对特征学习算法是否成功起到决定性作用- 甚至比选择非监督式学习算法更重要。事实上，当我们将这些参数调节至其极限，我们可以达到迄今最好的性能表现，并且比其他更复杂的模型表现更好。我们实验的最好结果是使用k-means聚类法，这是相当出人意料的。该算法广泛使用于计算机视觉算法，但是尚未在深度特征学习中被广泛采用。具体来说，我们在CIFAR-10数据集上测试正确率为79.6%，在NORB数据集上为97.2%，这优于以往发布的所有结果。

开始，我们将回顾特征学习的相关工作，接下来介绍通用的特征学习框架，我们将在第3节用它来测试并评价我们的系统。最后我们在第4节介绍我们的实验分析和展示针对CIFAR-10<sup>[5]</sup> 和NORB<sup>[4]</sup>数据集上的实验结果。

## 二、相关工作

自推出无监督训练前<sup>[2]</sup>，学者提出了许多新的方案，通过堆叠特征层次来获得“深层”表示。主要都集中在创建新的训练算法来建立单层模型以便组成深层模型。在历史

文献中，我们主要研究的算法是稀疏编码<sup>[6;8;9]</sup>，限制玻尔兹曼机<sup>[2;5]</sup>，稀疏限制玻尔兹曼机<sup>[10]</sup>，稀疏自动编码器<sup>[11;12]</sup>，去噪自动编码器<sup>[10]</sup>，分解<sup>[13]</sup>和平均协方差<sup>[14]</sup>限制玻尔兹曼机，以及其他算法<sup>[15;16]</sup>。因此，除了特征学习构架中的许多模块，非监督式学习模块是最重要的部分。

然而，一些工作已经研究了其他选择对特征学习系统所产生影响，尤其是不同网络架构产生的影响。例如，加勒特等人研究了改变特征层次之间的池化策略和在层之间采用不同的归一化和校正手段对系统产生的改变<sup>[1]</sup>。同样地，Boureau等人从理论<sup>[17]</sup>和实践<sup>[18]</sup>两方面研究了不同的编码策略和池化方法对系统产生的影响。我们的工作也遵循上述思路，但研究单层模型的结构，而不是池化方法，也不是不同算法和编码方案。

许多计算机视觉文献也涉及到我们的工作和涉及到特征学习算法。例如，我们将使用k-means聚类算法作为一种替代的非监督式学习算法模块。k-means在“深度学习”中使用不是很广泛，但在计算机视觉领域中是“视觉语言”的重要组成部分并被广泛采纳，他被用来刻画更高层次的图像特征<sup>[19-22]</sup>。这种方法也被递归地应用于构建多层特征<sup>[23]</sup>。池化手段和激励函数的选定或者编码方案的不同对系统产生的影响已经被研究并讨论<sup>[7;21;24]</sup>。例如，Van Gemert等人证明了“软”激励函数（“核函数”）比硬分配的激励函数在视觉词汇模型中产生的效果更好。

本文也会与前人的实验结果对照比较（例如，我们也会考虑“软”和“硬”激活函数），但是其他方面有一些不同：虽然我们可以证实，一些特征学习方案优于其他方案，我们仍展示这些所谓的不同通常因其他因素而被高估，例如特征的数量。因此，即使更复杂的学习策略可以略微优化性能，但这些可以被一些简洁但运算迅速的学习算法所替代，他们能处理更大的网络模型。

### 三、 非监督特征学习框架

在本节中，我们将描述一个用于特征学习的通用框架。具体来说，我们将集中于将这些算法应用到从图像中学习特征，尽管我们的算法也适用于其他形式的数据。我们使用的框架包含几个部分，这些部分与计算机视觉<sup>[19;21-24]</sup>和特征学习<sup>[4;15;17]</sup>领域算法框架十分相似。在较高的抽象层次，我们的系统将执行以下步骤来学习特征表示：

1. 从未标记训练图像中随机选取图像小块。
2. 对这些图像小块做预处理。
3. 利用非监督式学习算法学习特征映射函数。

当我们获取到特征映射函数和一系列的已标记的训练图像，我们将采用特征提取和分类：

1. 将输入图像划分成等间隔的小块，从中提取特征。
2. 池化将输入图像的不同区域，以减少特征数量。
3. 训练一个线性分类器，一预测给定特征向量的标签。

现在，我们就将更加详细地描述这套流程和其中的参数配置。

### 3.1 特征学习

如上所述，系统先随机从无标记输入图像中提取子块。每一个子块尺寸大小为 $w * w$ 并且有 $d$ 个通道， $w$ 代表“感受野的大小”。每一个 $w * w$ 的子块能重排列成 $R^N$ 维度的图像密度向量， $N = w * w * d$ 。然后我们构建 $m$ 个随机采样的小块， $X = \{x^{(1)}, \dots, x^{(m)}\}$ ，其中， $x^{(i)} \in R^N$ 。给定这样的数据集，我们接下来进行数据预处理和非监督式学习步骤。

#### 3.1.1 预处理

通常在从数据中获取特征之前，我们需要进行简单的归一化处理步骤。在这部分工作中，我们对每一个图像小块中的 $x^{(i)}$ 减去图像小块的平均值并除以其标准差，以实现归一化处理。针对可视化数据，这步对应于局部亮度和对比度归一化处理。

当输入向量做完归一化处理后，我们可以选择是否进行白化处理<sup>[25]</sup>。虽然这个处理手段经常被用于深度学习实验中（例如，<sup>[25]</sup>），但却在计算机视觉领域不常被用到。我们将对白化处理做对照实验，以决定这步骤是否是必须的。

#### 3.1.2 非监督式学习

在预处理之后，我们采用非监督式学习算法从无标签数据中发现特征。对我们而言，我们将非监督式学习算法看作是一个“黑匣子”，它读取输入数据 $X$ ，输出一个映射函数 $f: R^N \rightarrow R^K$ ，它将输入向量映射 $x^{(i)}$ 成 $K$ 个特征向量， $K$ 是算法的参数。我们将第 $k$ 个特征表示为 $f_k$ 。在这项工作中，我们将使用多种不同的非监督式学习算法：

稀疏自动编码器；

稀疏限制玻尔兹曼机；

**k-means聚类:**

**高斯混合模型。**

我们将简要总结如何将这此算法应用到我们的系统中去。

(1). **稀疏自动编码器:**我们训练具有 $K$ 个隐藏节点的自动编码器, 使用反向传播技术来缩减平方误差与附加惩罚项, 它能使节点保持一个较低平均激活值<sup>[3;11]</sup>。算法输出为权重矩阵 $W \in R^{K \times N}$ 和偏置项 $b \in R^K$ , 特征映射函数 $f$ 定义为:

$$f(x) = g(Wx + b) \quad (3.1)$$

训练算法中使用了许多超参数 (例如, 权重衰减项, 目标函数)。针对每一个感受野大小为 $w$ , 我们通过交叉验证手段来调节这些参数。

(2). **稀疏限制玻尔兹曼机:**限制玻尔兹曼机(RBM)是具有 $K$ 个二元隐含变量的无向图模型。稀疏RBM需要加上类似于自动编码器的稀疏惩罚因子, 可以利用差分逼近法<sup>[26]</sup>来训练。训练结果是权重矩阵 $W$ 和偏置项 $b$ , 从而获得类似于自动编码器的特征映射函数(公式3.1)——这样, 这些算法的差异主要是他们的训练手段不同而已。如上所述, 对于每一个不同大小的感受野尺寸, 我们也采用交叉验证手段加以调节。

(3). **k-means聚类:**我们应用K-means聚类方法到输入数据来学习 $K$ 个质心 $c^{(k)}$ 。当获取到质心 $c^{(k)}$ , 我们采用两种不同的特征映射函数 $f$ 。第一种是硬分配的编码策略:

$$f_k(x) = \begin{cases} 1 & \text{if } k = \arg \min_j \|c^{(j)} - x\|_2^2 \\ 0 & \text{otherwise.} \end{cases} \quad (3.2)$$

这种稀疏表示模式已经在计算机视觉中广泛应用<sup>[19]</sup>。然而, 学者认为这种编码方式可能太简单了<sup>[24]</sup>。因此我们的第二种特征映射函数是一个非线性函数, 它能比上述编码方案更加“适合”同时也能保持稀疏度:

$$f_k(x) = \max\{0, \mu(z) - z_k\} \quad (3.3)$$

其中,  $z_k = \|x - c^{(k)}\|$ ,  $\mu(z)$ 代表集合 $z$ 的平均值。对任意特征值 $f_k$ , 当它与其质心的距离 $c^k$ 距离超过平均值, 该激活函数将输出0。在实践中, 这意味着将近一半的特征值将被设置为0。这也可以作为特征值之间的一种简单的“竞争”。我们分别将这两种方案称为k-means (硬) 和k-means (三角)。

(4). **高斯混合模型:**高斯混合模型(GMMs)常用于聚类, 它将输入数据的密度表示成 $K$ 个高斯分布的聚合。该模型能利用最大期望算法 (EM) 来训练<sup>[23]</sup>。我们将迭代一遍

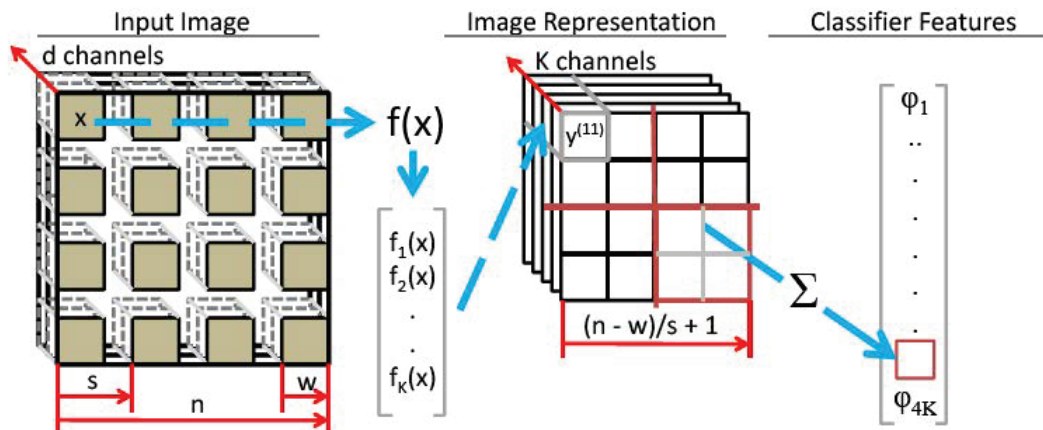


图 3.1 白化的效果和质心的数量

的k-means聚类的结果用来初始化该混合模型。映射函数 $f$ ，将输入映射成为其后验概率：

$$f_k(x) = \frac{1}{(2\pi)^{d/2} |\sum_k|^{1/2}} \exp\left(-\frac{1}{2}(x - c^{(k)})^T \sum_k^{-1} (x - c^{(k)})\right) \quad (3.4)$$

其中 $\sum_k$ 是对角协方差值， $\phi_k$ 是由EM算法学习得到的集群先验概率。

## 3.2 特征提取和分类

对特定的非监督式学习算法来说，上述步骤会产出一个函数 $f$ ，他会将输入小块 $x \in R^N$ 转换成新的表示形式 $y = f(x) \in R^K$ 。我们将利用这个特征提取器从我们有标签的训练图像中获取特征并进行分类。

### 3.2.1 卷积提取

对任意给定 $w \times w$ 大小的图像小块，利用学习到的特征提取器 $f: R^N \rightarrow R^K$ ，我们可以计算获得它新的表示形式 $y \in R^K$ 。通过该步骤，我们可以获得整个图像的单层表示形式。具体地说，给定一个 $n \times n$ 像素大小(有 $d$ 个通道)的图片，我们计算每一个大小为 $w \times w$ 的子块，获得了 $(n-w+1) \times (n-w+1)$ 大小的特征表示(有 $K$ 个通道)。形式上说，我们令 $y^{(ij)}$ 为输入图像中以坐标 $(i, j)$ 开始的大小为 $w \times w$ 的第 $K$ 通道的图像小块的新特征表示。考虑到计算上的效率，我们也可以调节特征提取器的步幅，令其大于1。在图3.1展示了上述实验步骤。

### 3.2.2 分类

在分类之前，池化方法能有效减少图像的维度。若步幅设置为 $s = 1$ ，我们通过特征映

射就能获得 $(n - w + 1) * (n - w + 1) * K$ 大小的特征表示。我们可以把同为 $y^{(ij)}$ 的区域都相加起来，以进一步缩减维度。这样的操作在计算机视觉领域<sup>[21]</sup>和深度学习领域<sup>[1]</sup>都广泛应用。在我们的系统中，我们采用十分简洁形式的池化方法。具体来说，我们将 $y^{(ij)}$ 划分成四个相等大小的象限。并分别计算每个 $y^{(ij)}$ 在其上的总和。对于每个象限，我们实现了维度规约，我们将数量为 $4K$ 的特征用来分类。

## 四、 实验与分析

上述框架包含多个可以调节的参数：

- 是否使用白化手段，
- 特征的数量 $K$ ，
- 步幅的大小 $s$ ，
- 感受野的大小 $w$ 。

在本节中，我们将展示着这参数对系统性能影响的实验结果。首先，我们将会用CIFAR-10训练数据集上利用交叉验证手段来评估这些参数的影响。然后，我们利用各自的非监督式学习算法，按照上述实验得出的分析结论并配置其最佳参数（即，在我们最后的结果中，对所有的算法我们将使用相同的设置），并给出在CIFAR-10和NORB数据测试集上面的实验结果。

我们基本的测试步骤如下。对于3.1.2节中介绍的每一种非监督式学习算法，在不同的参数 $K, s, w$ 下，我们将分别利用白化后的数据和原始数据进行训练单层的特征。接下来，如3.2.2节中描述，我们将会训练一个线性分类器并测试其在对照数据集（主要用于分析）和测试数据集（用于最后的结论）上的表现。

### 4.1 可视化

在我们展示分类结果之前，我们首先展示学习到的特征图像。由稀疏自动编码器，稀疏限制玻尔兹曼机，k-means聚类，高斯混合模型学习到的质心展示在图4.1，其中感受野的大小为8个像素。众所周知，自动编码器和限制玻尔兹曼机能学习到类似于Gabor滤波器的滤波效果。我们可以用白化后的数据和原始数据分别获得该结果。然而，这些图片也说明了相似的结果也可以用聚类算法获得。特别地，虽然对原始聚类产生的执行与<sup>[20;27]</sup>—



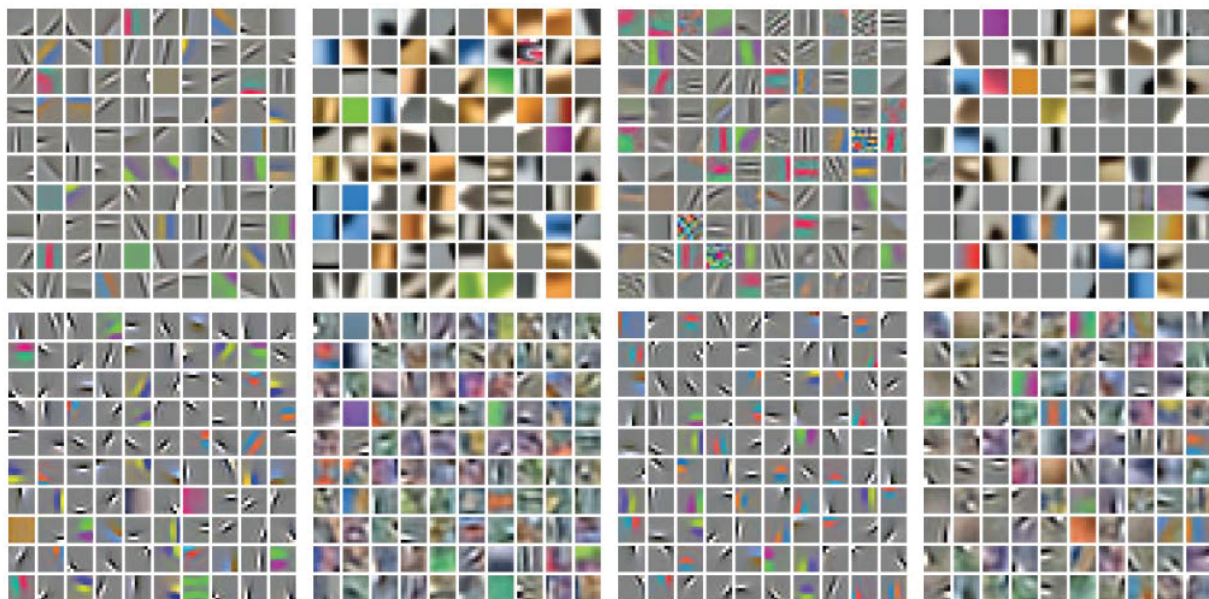


图 4.1 随机初始化质心并在CIFAR-10图像数据集上面应用不同学习算法训练结果。

致,但当我们对白化后的数据做聚类时,其效果远超过局部过滤器并十分类似于其他算法产生的结果。这样看来,对原始数据做白化处理,再使用聚类算法,我们可以轻易获得类似于其他算法的效果并且不需要调节参数。

## 4.2 白化的效果

接下来我们开始调整横轴的参数并研究其影响效果,首先是研究白化手段的影响效果,由图4.1可知,它能直接改变质心的位置。图4.2展示了所有算法在不同的特征数量和是否使用白化手段下它的总体表现情况,其中固定步幅为1,感受野大小为6个像素。

对于稀疏自动编码器和限制玻尔兹曼机,白化手段的效果不是很明显。当只使用100特征数量时,稀疏限制玻尔兹曼机的性能提升最明显,但当使用更多的特征数时,这种性能提升逐渐消失了。然而,对于聚类算法,我们发现白化手段是必不可少的预处理过程,因为聚类算法无法处理数据内部的相关性问题。

聚类算法已经成功应用在原始像素输入数据上面<sup>[20;27]</sup>,但这些应用尚未使用白化处理的数据数据。我们的结果表明如果使用白化手段,会有效提高它们的性能表现。

## 4.3 特征的数量

我们在实验中采用特征数量分别为100, 200, 400, 800, 1200和1600来学习特征表

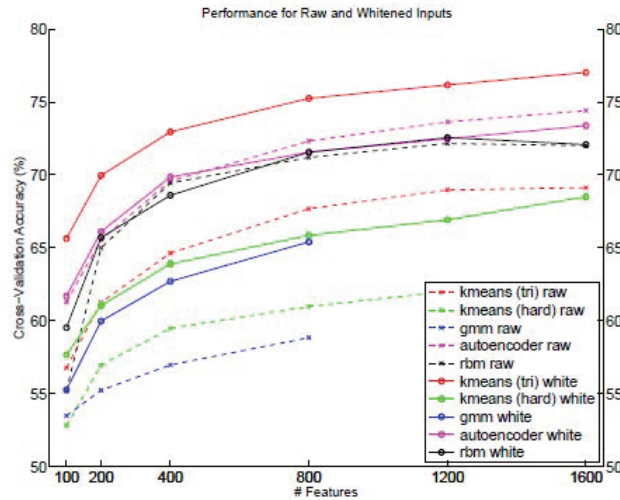


图 4.2 插图中显示了特征提取过程，其中感受野大小为  $w * w$ ，步幅为  $s$ 。我们首先提取  $w * w$  大小间隔为  $s$  像素的图像小块，然后将它们映射到  $K$  维特征向量，形成新的图像特征表示。这些向量接着被池化并被划分成四个象限，最后拼接成特征向量以便于分类工作。（为了清楚起见，我们取右边的图时步幅大于  $w$ ，但实际上平均步幅大小均小于  $w$ ）。

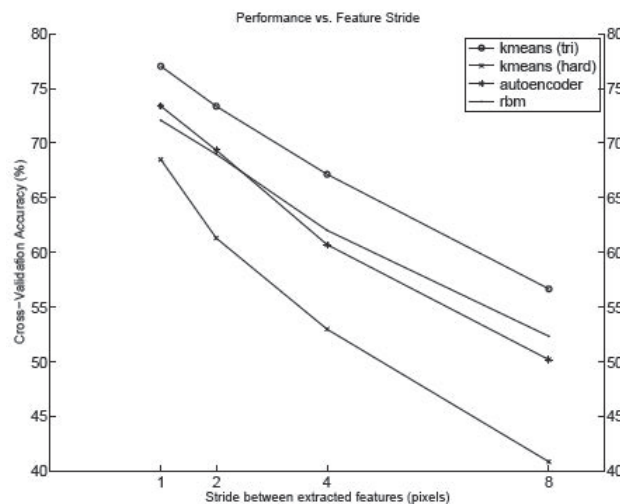


图 4.3 步幅的效果

示。图4.3清楚地显示了增加特征数量后带来的效果：所有的算法都随着特征数量的增加而性能提高。

出人意料的是，采用k-means聚类算法，“三角”激活函数和白化处理手段时系统的性能表现最好。特别令人关注的是，k-means聚类无需调节任何参数，不像其他的稀疏自动编码器和稀疏玻尔兹曼机，他们需要调节很多超参数才能达到最好的实验结果。

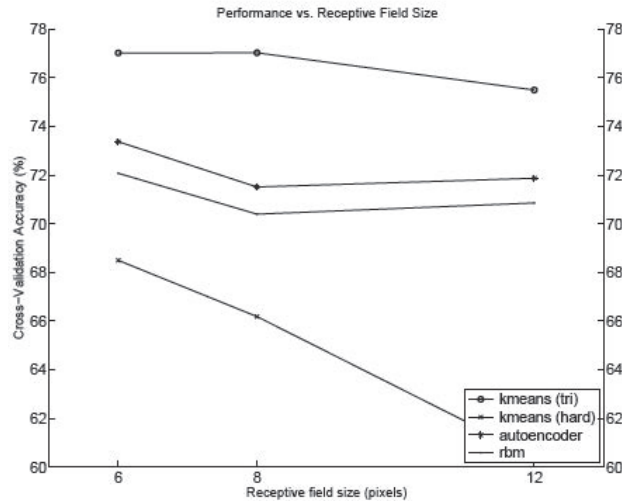


图 4.4 感受野尺寸的影响

#### 4.4 步幅的影响

在我们的框架中，“步幅” $s$ 指的是提取特征值时两个图像小块之间的距离（参见图3.1）。通常，学习系统的步幅 $s > 1$ ，因为计算特征映射十分耗费计算资源。例如，稀疏编码要求对每一个图像小块求解优化问题的解，这样把步幅设置为1，显然是不合理的。那么我们想要了解为了妥协性能方面的因素，我们会牺牲多少系统性能（他们都有自己的特征映射函数，可以非常迅速地计算出他们的特征值）。在本实验中，我们设置特征数量为1600，感受野的大小为6像素，却改变步幅的大小：1，2，4和8像素。实验结果展示在图4.4中（我们不做高斯混合模型的实验，因为这样的参数配置对它来说不合理）。

图中显示随着步长的增大，性能有明显的下降。然而，变化的幅度十分惊人：即使步幅设置为 $s = 2$ ，我们损失了3%的精确度。当 $s = 4$ 时我们损失了5%。在比较算法间的差异时，这种误差更加明显。例如，我们将稀疏限制玻尔兹曼机算法的步幅设置为 $s = 2$ 它的性能表现与硬分配的k-means聚类算法步幅设置为 $s = 1$ 时相同——该算法是在非监督算法领域最简单的算法（显然比稀疏限制玻尔兹曼机算法简单很多）。

#### 4.5 感受野的大小

最后，我们还评估了感受野的大小产生的影响。给定一个可扩展的算法，若是扩大感受野的大小，他能学习到更复杂的特征，能涵盖图像更大的区域。另一方面，随着空间维度地增加，我们的算法需要学习更多的特征，需要更多的数据作支撑。因此，在相同的数据量和相同的特征数量下，我们尚未确定这是一个有效的措施。在本实验中，我们设置感

受野的大小为6, 8, 12像素。对于其他参数, 我们使用白化手段, 并且步幅为1像素, 质心数量为1600。

实验结果展示在图<sup>[19]</sup>。总体来说, 感受野大小为6像素时效果最好, 12像素大小的感受野产生的效果比其余情况都要差。因此, 如果我们还有空余的计算资源, 我们的实验结果表明, 最好能将它用来减少步幅 $s$ 并增加学习特征的数量。

不幸的是, 不像其他参数, 感受野的尺寸需要采用交叉验证手段来做出明智的选择。我们的实验表明, 感受野的尺寸较小时, 它可以表现的很好。这一点尤其重要, 特别是当我们降低输入尺寸以便使用更小的步幅和学习更多的特征, 他们都能对结果有明显的提升。

## 4.6 最后的分类结果

实验表明, 针对CIFAR-10数据集, 采用白化手段, 步幅设置为1像素, 感受野设置为6像素, 采用大量的特征能在上诉算法中获得最好的实验结果。我们使用上述参数配置, 在整个CIFAR-10训练集上面完成全部流程, 再用来训练支持向量机分类器并在标准的CIFAR-10测试数据集上面做测试。我们最后的结果展示在表格4.1中, 表格还包含了其他文献中的实验结果。很奇怪的是, 当特征数量为1600时, k-means(三角)算法达到了非常高的性能(77.9%)。在此基础之上, 我们试图通过增加特征数来那个来获得更好的效果, 果不其然, 当特征数量为4000时, 我们的实验精度为79.6%。

基于上述分析, 我们在NORB数据集(数据已归一化)上运行了所有的算法。我们使用CIFAR-10数据集上获得的参数设置, 包括感受野的大小为6像素。实验结果总结在表格4.2中。其中, 所有的算法都能实现非常高的性能。同样, K-means聚类算法并使用“三角”激活函数能实现最高的性能。当我们设置CIFAR数据集上的特征数量为4000时, 我们的实验正确率达到了97.21%。然而, 我们注意到其他算法获得的结果十分相似, 无论使用什么算法。这表明, 系统性能主要的因素是网络架构而不是选择非监督式学习算法的不同。

最后, 我们在STL-10数据集上运行了我们的系统。该数据集使用更高的分辨率(96\*96)的图像, 但训练样本很少(每个类别是由100个实例), 但却提供了大量未标记的训练集——从而迫使算法从图像中统计获得先验知识。我们对STL图像进行降采样(32\*32像素)并使用与CIFAR数据集相同的实验系统。在这种情况下, 其性能表现远比CIFAR上的小的已标记的数据集低很多: 51.5%( $\pm 1.73\%$ )(若采用原始数据, 31.8%( $\pm 0.62\%$ ))。这表明, 当我们有大量标注训练集是像NORB和CIFAR数据集, 这里提出的方法是最佳选择。

表 4.1 CIFAR-10 数据集上的测试识别准确率

原始像素 <sup>[15]</sup> 介绍过)	37.3%
均值-方差玻尔兹曼机(3层) <sup>[14]</sup>	71.0%
卷积深度信念网络(2层) <sup>[28]</sup>	<b>78.9%</b>
稀疏自动编码器	73.4%
稀疏限制玻尔兹曼机	73.4%
K-means(硬)	68.6%
K-means(三角)	77.9%
K-means(三角, 4000特征数量)	<b>79.6%</b>

表 4.2 NORB 数据集上的测试识别准确率(和误差率)

算法	正确率(误差率)
卷积神经网络 <sup>[4]</sup>	93.4%(6.6%)
深度玻尔兹曼机 <sup>[29]</sup>	92.8%(7.2%)
深度信念网络 <sup>[30]</sup>	95.0%(5.0%)
深度神经网络 <sup>[31]</sup>	<b>97.13%(2.87%)</b>
稀疏自动编码器	96.9%(3.1%)
稀疏限制玻尔兹曼机	96.2%(3.8%)
K-means(硬)	96.9%(3.1%)
K-means(三角)	97.0%(3.0%)
K-means(三角, 4000特征数量)	<b>97.21%(2.79%)</b>

表 4.3 STL-10 数据集上的测试识别准确率

算法	正确率(误差率)
原始像素	31.8%(± 0.62%)
K-means(三角, 1600特征数量)	<b>51.5%(± 1.73%)</b>

## 五、 讨论

考虑到我们实验系统的简单性，上面的结果看起来莫名其妙——当我们第一次检查实验结果时，我们尚未明确什么因素导致实验结果远高于先前的工作的结果。我们认为这样的性能增益主要解释是，主要是我们合理配置了网络参数，由于所有算法的结果都比以前的执行结果更好。针对每一个网络结构的参数(特征数量，步幅和感受野的大小)，我们都对其做对照实验分析其对性能提升的效果。例如，使用大量的特征(无论他们是通过何种训练方式获得的)为我们提供了数据的非线性映射方式。不同于简单具有有限表达能力的线性映射，当我们使用更多的非线性映射函数，我们可以使数据线性可分以便以分类。因此，不管训练算法如何，使用大量的特征数量我们可以一致的提升系统性能。

通过改变步幅大小能显著改变性能，Boureau也分析了部分原因<sup>[18]</sup>。通过设置较小的步幅，大量的采样数据被融入池化后的数据，在理论和以往的经验中，这样的操作能改善实验结果。高频特征(边缘)很可能是使用密集的采样才能更准确识别的。

最后，通过交叉验证选择的感受野尺寸大小也是至关重要的因素。实验表明，采用大尺寸的感受野区域能用少量小尺寸的感受野区域非线性的覆盖。例如，由于我们的特征通常包含边缘位移的副本，增加尺寸的同时也增加了冗余度。这样的情况可以通过使用卷积手段来训练<sup>[4;15;32]</sup>。需要注意的是，小尺寸的感受野区域也可能增加池化后的样本个数从而它的效果类似于采用步幅小的系统。

## 六、 总结

在本文中，我们在CIFAR-10数据集上使用多种无监督学习算法做了大量实验，来刻画各种参数对分类效果的影响。我们不只一次确认了特征数量越多，提取越密集，有助于优化系统性能，而且我们发现这些配置参数在实验中与无监督学习算法一样重要。出人意料的是，我们发现K-means聚类算法——一个十分简单的不需要调节参数的算法——在使用我们在实验中发现的最佳参数配置后，能在CIFAR-10和NORB数据集上实现最好的实验结果。更一般地，步幅越小，特征数量越多能有效提高系统精确度。这表明，虽然复杂的算法能获得更强的特征表示能力，但简洁、运算迅速的算法也是相当有竞争能力的。

## 参考文献

- [1] Kevin Jarrett, Koray Kavukcuoglu, M Ranzato, Yann LeCun. What is the best multi-stage architecture for object recognition?[C]//Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009:2146–2153.
- [2] Geoffrey Hinton, Simon Osindero, Yee-Whye Teh. A fast learning algorithm for deep belief nets[J]. Neural computation, 2006. 18(7):1527–1554.
- [3] 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.
- [4] Yann LeCun, Fu Jie Huang, Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting[C]//Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. volume 2. IEEE, 2004:II–97.
- [5] Alex Krizhevsky, Geoffrey Hinton. Learning multiple layers of features from tiny images[J]. Computer Science Department, University of Toronto, Tech. Rep, 2009. 1(4):7.
- [6] Jianchao Yang, Kai Yu, Yihong Gong, Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification[C]//Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009:1794–1801.
- [7] Eric Nowak, Frédéric Jurie, Bill Triggs. Sampling strategies for bag-of-features image classification. //Computer Vision–ECCV 2006. Springer, 2006. 490–503.
- [8] 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.
- [9] Honglak Lee, Alexis Battle, Rajat Raina, Andrew Y Ng. Efficient sparse coding algorithms[C]//Advances in neural information processing systems. 2006:801–808.
- [10] 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.
- [11] Ian Goodfellow, Honglak Lee, Quoc V Le, Andrew Saxe, Andrew Y Ng. Measuring invariances in deep networks[C]//Advances in neural information processing systems. 2009:646–654.
- [12] Christopher Poultney, Sumit Chopra, Yann L Cun, et al. Efficient learning of sparse representations with an energy-based model[C]//Advances in neural information processing systems. 2006:1137–1144.
- [13] Alex Krizhevsky, Geoffrey E Hinton, et al. Factored 3-way restricted boltzmann machines for modeling natural images[C]//International Conference on Artificial Intelligence and Statistics. 2010:621–628.
- [14] M Ranzato, Geoffrey E Hinton. Modeling pixel means and covariances using factorized third-order boltz-



- 
- mann machines[C]//Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010:2551–2558.
- [15] Honglak Lee, Roger Grosse, Rajesh Ranganath, Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations[C]//Proceedings of the 26th Annual International Conference on Machine Learning. ACM, 2009:609–616.
- [16] Kai Yu, Tong Zhang. Improved local coordinate coding using local tangents[C]//Proceedings of the 27th International Conference on Machine Learning (ICML-10). 2010:1215–1222.
- [17] Y-L Boureau, Francis Bach, Yann LeCun, Jean Ponce. Learning mid-level features for recognition[C]//Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010:2559–2566.
- [18] Y-Lan Boureau, Jean Ponce, Yann LeCun. A theoretical analysis of feature pooling in visual recognition[C]//Proceedings of the 27th International Conference on Machine Learning (ICML-10). 2010:111–118.
- [19] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, Cédric Bray. Visual categorization with bags of keypoints[C]//Workshop on statistical learning in computer vision, ECCV. volume 1. Prague, 2004:1–2.
- [20] Li Fei-Fei, Pietro Perona. A bayesian hierarchical model for learning natural scene categories[C]//Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. volume 2. IEEE, 2005:524–531.
- [21] Svetlana Lazebnik, Cordelia Schmid, Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories[C]//Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. volume 2. IEEE, 2006:2169–2178.
- [22] John Winn, Antonio Criminisi, Thomas Minka. Object categorization by learned universal visual dictionary[C]//Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. volume 2. IEEE, 2005:1800–1807.
- [23] Ankur Agarwal, Bill Triggs. Hyperfeatures–multilevel local coding for visual recognition. //Computer Vision–ECCV 2006. Springer, 2006. 30–43.
- [24] Jan C van Gemert, Jan-Mark Geusebroek, Cor J Veenman, Arnold WM Smeulders. Kernel codebooks for scene categorization. //Computer Vision–ECCV 2008. Springer, 2008. 696–709.
- [25] Aapo Hyvärinen, Erkki Oja. Independent component analysis: algorithms and applications[J]. Neural networks, 2000. 13(4):411–430.
- [26] Geoffrey Hinton. Training products of experts by minimizing contrastive divergence[J]. Neural computation, 2002. 14(8):1771–1800.
- [27] Manik Varma, Andrew Zisserman. A statistical approach to material classification using image patch ex-



- emplars[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2009. 31(11):2032–2047.
- [28] Alex Krizhevsky, G Hinton. Convolutional deep belief networks on cifar-10[J]. Unpublished manuscript, 2010.
- [29] Ruslan Salakhutdinov, Geoffrey E Hinton. Deep boltzmann machines[C]//International Conference on Artificial Intelligence and Statistics. 2009:448–455.
- [30] Vinod Nair, Geoffrey E Hinton. 3d object recognition with deep belief nets[C]//Advances in Neural Information Processing Systems. 2009:1339–1347.
- [31] Rafael Uetz, Sven Behnke. Large-scale object recognition with cuda-accelerated hierarchical neural networks[C]//Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on. volume 1. IEEE, 2009:536–541.
- [32] Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, Yann L Cun. Learning convolutional feature hierarchies for visual recognition[C]//Advances in neural information processing systems. 2010:1090–1098.