



(12) 发明专利申请

(10) 申请公布号 CN 112020724 A

(43) 申请公布日 2020.12.01

(21) 申请号 201980004446.2

(74) 专利代理机构 北京市柳沈律师事务所  
11105

(22) 申请日 2019.04.01

代理人 金玉洁

(85) PCT国际申请进入国家阶段日  
2020.03.10

(51) Int.Cl.

(86) PCT国际申请的申请数据  
PCT/US2019/025210 2019.04.01

G06N 3/04 (2006.01)

G06N 3/08 (2006.01)

(87) PCT国际申请的公布数据  
W02020/204904 EN 2020.10.08

(71) 申请人 谷歌有限责任公司  
地址 美国加利福尼亚州

(72) 发明人 A.施里瓦斯塔瓦 S.辛格 J.巴尔  
S.A.阿布-埃尔-哈贾 N.约翰斯顿  
G.D.托德里西

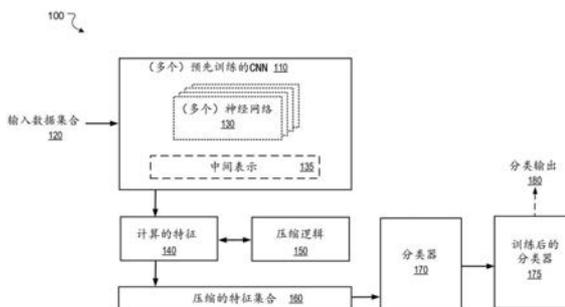
权利要求书4页 说明书14页 附图8页

(54) 发明名称

学习可压缩的特征

(57) 摘要

包括在计算机存储介质上编码的计算机程序的方法、系统和装置,用于通过神经网络(NN)接收数据集合以用于根据数据集合生成特征。至少使用NN的特征层根据数据集合来计算第一特征集合。第一特征集合i)由信息量的度量表征;并且ii)被计算以使得第一特征集合的尺寸可压缩为第二特征集合,第二特征集合在尺寸上小于第一特征集合并且具有与第一特征集合的信息量的度量相同的信息量的度量。第二特征集合是使用压缩第一特征集合以生成第二特征集合的压缩方法根据第一特征集合而生成的。



1. 一种用于计算特征的计算机实施的方法,所述方法包括:

由神经网络接收数据集合以用于生成从所述数据集合推导出的特征,其中,所述神经网络包括多个层,并且至少一个层是特征层;

使用神经网络的至少一个特征层,根据所述数据集合来计算第一特征集合,其中,第一特征集合:

i) 由信息量的度量表征;并且

ii) 被计算以使得第一特征集合的尺寸可压缩为第二特征集合,第二特征集合得到在尺寸上小于第一特征集合的尺寸的特征集合,并且具有与第一特征集合的信息量的度量相同的信息量的度量;以及

使用压缩第一特征集合以生成第二特征集合的压缩方法,根据第一特征集合来生成第二特征集合。

2. 根据权利要求1所述的方法,其中,生成第二特征集合包括:

将特征层识别为神经网络的瓶颈层,以用于降低使用神经网络计算的特征的维数;

使用在瓶颈层处计算的特征来生成中间表示,其中,所述中间表示是基于损失函数而生成的;以及

响应于将所述压缩方法应用于根据中间表示而计算的第一特征集合,生成第二特征集合。

3. 根据权利要求1所述的方法,其中,生成所述中间表示包括:

将熵项应用于损失函数以获得增强的损失函数;以及

通过使用增强的损失函数计算神经网络的瓶颈层处的特征来生成中间表示。

4. 根据权利要求3所述的方法,其中,所述损失函数包括量化元素,并且将所述熵项应用于所述损失函数以获得所述增强的损失函数包括:

在硬件电路上实施的神经网络的训练阶段期间,用加性均匀噪声元素替代损失函数的量化元素。

5. 根据权利要求3所述的方法,其中,所述中间表示的计算的特征具有受所述熵项影响的压缩率的度量,所述熵项被应用以获得所述增强的损失函数。

6. 根据权利要求1所述的方法,其中,计算第一特征集合包括:

使用神经网络的特征层之前的神经网络的一个或多个层来计算第一特征集合的至少一部分。

7. 根据权利要求1所述的方法,其中,所述瓶颈层用于降低计算的特征的维数,而不削弱计算的特征的预测能力。

8. 根据权利要求1所述的方法,其中,所述数据集合包括图像数据,并且生成第二特征集合包括:

训练神经网络以学习指示所述数据集合的图像数据的特性的特征的层次结构,其中,所述特征的层次结构从包括图像中描绘的项目的线和边的低级图像特征到与所述数据集合的对象相关的高级语义特征。

9. 根据权利要求1所述的方法,其中,第二特征集合是用于训练分类器的数据,并且所述方法还包括:

训练神经网络分类器,以基于第二特征集合中的特征的特性来检测所述数据集合中的

数据对象的特性,其中,通过经由所述神经网络分类器的一个或多个层处理第二特征集合来训练所述神经网络分类器。

10. 根据权利要求1所述的方法,其中,第二特征集合是表示第一特征集合的压缩版本的压缩的特征集合。

11. 根据权利要求11所述的方法,其中,第一特征集合的信息量由以下中的至少一个来定义:

第一特征集合中的计算的信息的精度的度量;或者

第一特征集合中的计算的信息的维数。

12. 根据权利要求1所述的方法,其中,所述神经网络是预先训练的卷积神经网络,其被配置为特征生成器以执行与自动化特征工程相关的一个或多个函数。

13. 根据权利要求1所述的方法,其中,所述神经网络:

在硬件电路上实施;并且

是前馈特征检测器网络。

14. 一种用于计算特征的系统,所述系统包括:

一个或多个处理设备;以及

一个或多个非暂时性机器可读存储设备,其存储可由一个或多个处理设备执行以执行操作的指令,所述操作包括:

由神经网络接收数据集合,以用于生成从所述数据集合推导出的特征,其中,所述神经网络包括多个层,并且至少一个层是特征层;

使用神经网络的至少一个特征层,根据所述数据集合来计算第一特征集合,其中,第一特征集合:

i) 由信息量的度量表征;并且

ii) 被计算以使得第一特征集合的尺寸可压缩为第二特征集合,第二特征集合得到在尺寸上小于第一特征集合的尺寸的特征集合,并且具有与第一特征集合的信息量的度量相同的信息量的度量;以及

使用压缩第一特征集合以生成第二特征集合的压缩方法,根据第一特征集合来生成第二特征集合。

15. 根据权利要求14所述的系统,其中,生成第二特征集合包括:

将特征层识别为神经网络的瓶颈层,以用于降低使用神经网络计算的特征的维数;

使用在瓶颈层处计算的特征生成中间表示,其中,所述中间表示是基于损失函数而生成的;以及

响应于将所述压缩方法应用于根据中间表示而计算的第一特征集合,生成第二特征集合。

16. 根据权利要求14所述的系统,其中,生成所述中间表示包括:

将熵项应用于损失函数以获得增强的损失函数;以及

通过使用增强的损失函数计算神经网络的瓶颈层处的特征来生成中间表示。

17. 根据权利要求16所述的系统,其中,所述损失函数包括量化元素,并且将所述熵项应用于所述损失函数以获得所述增强的损失函数包括:

在硬件电路上实施的神经网络的训练阶段期间,用加性均匀噪声元素替代损失函数的

量化元素。

18. 根据权利要求16所述的系统,其中,所述中间表示的计算的特征具有受所述熵项影响的压缩率的度量,所述熵项被应用以获得所述增强的损失函数。

19. 根据权利要求14所述的系统,其中,计算第一特征集合包括:

使用神经网络的特征层之前的神经网络的一个或多个层来计算第一特征集合的至少一部分。

20. 根据权利要求14所述的系统,其中,所述瓶颈层用于降低计算的特征的维数,而不削弱计算的特征的预测能力。

21. 根据权利要求14所述的系统,其中,所述数据集合包括图像数据,并且生成第二特征集合包括:

训练神经网络以学习指示所述数据集合的图像数据的特性的特征的层次结构,其中,所述特征的层次结构从包括图像中描绘的项目的线和边的低级图像特征到与所述数据集合的对象相关的高级语义特征。

22. 根据权利要求14所述的系统,其中,第二特征集合是用于训练分类器的数据,并且所述操作还包括:

训练神经网络分类器以基于第二特征集合中的特征的特性来检测所述数据集合中的数据对象的特性,其中,通过经由所述神经网络分类器的一个或多个层处理第二特征集合来训练所述神经网络分类器。

23. 根据权利要求14所述的系统,其中,第二特征集合是表示第一特征集合的压缩版本的压缩的特征集合。

24. 根据权利要求23所述的系统,其中,第一特征集合的信息量由以下中的至少一个来定义:

第一特征集合中的计算的信息的精度的度量;或者

第一特征集合中的计算的信息的维数。

25. 根据权利要求14所述的系统,其中,在所述硬件电路上实施的神经网络是预先训练的卷积神经网络,其被配置为特征生成器以执行与自动化特征工程相关的一个或多个函数。

26. 根据权利要求14所述的系统,其中,所述神经网络:

在硬件电路上实施;并且

是前馈特征检测器网络。

27. 一个或多个非暂时性机器可读存储设备,其存储可由一个或多个处理设备执行以执行操作的指令,所述操作包括:

由神经网络接收数据集合以用于生成从所述数据集合推导出的特征,其中,所述神经网络包括多个层,并且至少一个层是特征层;

使用神经网络的至少一个特征层,根据所述数据集合来计算第一特征集合,其中,第一特征集合:

i) 由信息量的度量表征;并且

ii) 被计算以使得第一特征集合的尺寸可压缩为第二特征集合,第二特征集合得到在尺寸上小于第一特征集合的尺寸的特征集合,并且具有与第一特征集合的信息量的度量相

同的信息量的度量;以及

使用压缩第一特征集合以生成第二特征集合的压缩方法,根据第一特征集合来生成第二特征集合。

## 学习可压缩的特征

### 背景技术

[0001] 本说明书涉及使用神经网络来生成特征集合。

[0002] 神经网络是采用一个或多个层的操作针对接收到的输入生成输出(例如,分类)的机器学习模型。一些神经网络除了输出层之外还包括一个或多个隐藏层。每个隐藏层的输出被用作网络中下一层(即网络的下一隐藏层或输出层)的输入。网络的一些层或所有层根据相应参数集合的当前值从接收到的输入中生成输出。

[0003] 一些神经网络包括一个或多个卷积神经网络(convolutional neural network, CNN)层。每个卷积神经网络层具有相关联的内核集合。每个内核包括由用户创建的神经网络模型所建立的值。在一些实施方式中,内核识别特定的图像轮廓、形状或颜色。内核可以被表示为权重输入的矩阵结构。每个卷积层也可以处理激活输入集合。激活输入集合也可以被表示为矩阵结构。

### 发明内容

[0004] 对于机器学习,特征学习是允许系统从原始数据中自动地发现特征检测或分类所需的表示的技术集合。特征学习可以是自动化过程(例如,代替手动特征工程),其允许机器来学习特征集合以及使用特征执行特定任务。在一些示例中,特定任务可以涉及训练分类器(诸如神经网络分类器)以检测项目或文档的特性。

[0005] 特征通常是由将进行分析或预测的独立单元共享的属性或性质。例如,独立单元可以是形成项目(诸如图像和其他文档)的部分的图像像素组。特征可以是图像中描绘的对象的属性,诸如由图像像素组定义的线或边。通常,任何属性都可以是特征,只要该属性对于执行模型的期望的分类函数是有用的。因此,对于给定的问题,当解决该问题时,特别是当解决该问题涉及进行关于可能有帮助的数据集合的一些预测时,特征可以是该数据集中的特性。

[0006] 示例人工神经网络(“NN(neural network)”)可以被预先训练以执行一个或多个数据分析函数。预先训练的NN可以具有高维的且存储成本高的特征集合或特征向量集合。用于高效地存储这些特征的常规方法会导致失真,失真降低了表示特征的数据的质量和有效性。特征可以由神经网络使用以执行与例如图像识别或对象识别相关的各种函数。在一些实施方式中,对于靶向任务随时间演变的某些类型的应用,可能需要预先计算和存储特征。

[0007] 预先训练的CNN作为现成的特征生成器非常强大,并且已经显示出对各种任务都表现得非常好。不幸的是,所生成的特征是高维的并且存储昂贵。例如,存储所生成的特征可能需要每示例潜在地存储成千上万个浮点数,特别是当处理视频文件的电子数据时。传统的基于熵的压缩方法没什么帮助,并且通用的有损替代方法是次优的。

[0008] 本文档描述了与用于学习特征的原始靶向目标一起为原始特征集合的高效压缩率(compressibility)进行联合优化的机器学习方法。该方法包括将特征层识别为瓶颈,在该瓶颈处,靶向目标鼓励信息流并与压缩率的目的背道而驰。所描述的方法引入了使得能

能够在特征的压缩率和信息量 (informativeness) 之间进行权衡的惩罚 (penalty)。该方法插件 (plug-in) 本质使其可以直接与任何靶向目标相结合。本文档中的描述证明, 基于这些方法而学习到的特性, 在保持其对靶向任务的信息量的同时, 其可压缩性与传统的压缩方法相比提高了几个数量级。

[0009] 该文档还描述了训练神经网络的改进方法。所描述的技术可以产生期望的效果, 其中训练后的神经网络的瓶颈层的节点 (例如, 系数或权重) 适于使用常规压缩算法进行压缩, 训练后的神经网络代表已经由神经网络学习的输入数据的特征的泛化, 常规压缩算法通常受益于要压缩的数据的很低的熵。例如, 为了实现这种技术效果, 这些技术包括用熵项 (entropy term) 增强损失函数。熵项可以对应于针对瓶颈层的系数/权重的概率模型, 其中概率模型可以通过与至少一个其他神经网络联合训练的单独的神经网络来实施。

[0010] 本说明书中描述的主题的一个方面可以体现在用于计算特征以用于训练和测试机器学习模型 (诸如分类器) 的计算机实施的方法中。该方法包括由神经网络接收数据集合以用于生成从数据集合推导出的特征。神经网络包括多个层, 并且至少一个层是特征层。该方法包括至少使用神经网络的特征层, 根据数据集合来计算第一特征集合。第一特征集合: i) 由信息量的度量表征; 并且 ii) 被计算以使得第一特征集合的尺寸可压缩为第二特征集合, 第二特征集合得到在尺寸上小于第一特征集合的尺寸的特征集合, 并且具有与第一特征集合的信息量的度量相同的信息量的度量。该方法还包括使用压缩第一特征集合以生成第二特征集合的压缩方法, 根据第一特征集合来生成第二特征集合。

[0011] 这些和其他实施方式可以各自可选地包括以下特征中的一个或多个。例如, 在一些实施方式中, 生成第二特征集合包括: 将特征层识别为神经网络的瓶颈层, 以用于降低使用神经网络计算的特征的维数; 使用在瓶颈层处计算的特征来生成中间表示, 其中该中间表示是基于损失函数而生成的; 以及响应于将压缩方法应用于根据中间表示而计算的第一特征集合, 生成第二特征集合。

[0012] 在一些实施方式中, 生成中间表示包括: 将熵项应用于损失函数以获得增强的损失函数; 以及通过使用增强的损失函数计算神经网络的瓶颈层处的特征来生成中间表示。在一些实施方式中, 损失函数包括量化元素, 并且将熵项应用于损失函数以获得增强的损失函数包括: 在硬件电路上实施的神经网络的训练阶段期间, 用加性均匀噪声元素替代损失函数的量化元素。

[0013] 在一些实施方式中, 计算的中间表示的特征具有受熵项影响的压缩率的度量, 熵项被应用以获得增强的损失函数。在一些实施方式中, 计算第一特征集合包括: 使用神经网络的特征层之前的神经网络的一个或多个层来计算第一特征集合的至少一部分。

[0014] 在一些实施方式中, 瓶颈层用于降低计算的特征的维数, 而不削弱计算的特征的预测能力。在一些实施方式中, 数据集合包括图像数据, 并且生成第二特征集合包括: 训练神经网络以学习指示数据集合的图像数据的特性的特征的层次结构 (hierarchy), 其中, 该特征的层次结构从包括图像中描绘的项目的线和边的低级图像特征到与数据集合的对象相关的高级语义特征。

[0015] 在一些实施方式中, 第二特征集合是用于训练分类器的数据, 并且该方法还包括: 训练神经网络分类器以基于第二特征集合中的特征的特性来检测数据集合中的数据对象的特性, 其中通过经由神经网络分类器的一个或多个层处理第二特征集合来训练神经网络

分类器。

[0016] 在一些实施方式中,第二特征集合是表示第一特征集合的压缩版本的压缩的特征集合。在一些实施方式中,第一特征集合的信息量由以下中的至少一个来定义:第一特征集合中的计算的信息的精度的度量;或者第一特征集合中的计算的信息的维数。

[0017] 在一些实施方式中,神经网络是预先训练的卷积神经网络,其被配置为特征生成器以执行与自动化特征工程相关的一个或多个函数。在一些实施方式中,神经网络:在硬件电路上实施;并且是前馈特征检测器网络。

[0018] 这个和其他方面的其他实施方式包括相对应的系统、装置和计算机程序,其被配置为执行方法的动作,被编码在计算机存储设备上。一个或多个计算机或电路的计算系统可以凭借安装在系统上的软件、固件、硬件或它们的组合来配置,使得在操作中使系统执行动作。一个或多个计算机程序可以凭借具有指令来配置,使得该指令在由数据处理设备执行时使设备执行动作。

[0019] 本说明书中描述的主题可以在特定实施例中实施,以实现以下优点中的一个或多个。所描述的方法允许根据原始输入数据集合来计算初始特征集合。计算的特征集合具有与特征集合的预测能力相对应的精度或信息量的特定度量。所描述的方法包括压缩该初始特征集合以生成具有与未压缩的特征集合相同或基本相同的精度或信息量的压缩的特征集合。计算的特征集合可以被确定尺寸以使得在该集合中存储计算的特征是禁止的或者需要大量的存储和计算资源。因此,所描述的方法可以用于计算特征集合,其中根据计算的特征而生成的压缩的特征集合保持其固有的精度和信息量,但是需要较少的内存和计算资源用于存储和检索。

[0020] 在本说明书中描述的主题的一个或多个实施方式的细节在附图和下面的描述中阐述。本主题的其他潜在特征、方面和优点将从说明书、附图和权利要求中变得显而易见。

## 附图说明

[0021] 图1示出了用于生成一个或多个特征集合的示例计算系统。

[0022] 图2A示出了使用图1的计算系统来生成压缩的特征集合的示例过程。

[0023] 图2B示出了与用于生成和评估压缩的特征集合的机器学习方法相关联的示例框图。

[0024] 图3A和图3B各自示出了关于与根据相应的标记的数据集合而生成的压缩的特征集合相关联的分类误差的图形数据。

[0025] 图4示出了关于与根据特定类型的标记的数据集合而生成的压缩的特征集合相关联的分类误差的示例图形数据。

[0026] 图5A示出了关于与使用两层神经网络生成的压缩的特征集合相关联的分类误差的图形数据。

[0027] 图5B示出了关于与使用三层神经网络生成的压缩的特征集合相关联的分类误差的图形数据。

[0028] 不同附图中相同的附图标记和指定表示相同的元件。

## 具体实施方式

[0029] 卷积神经网络 (CNN) 已经在计算机视觉和机器学习方面取得了成功, 并且帮助推动了各种问题上的前沿。CNN的成功归因于它们学习从非常低级的图像特征 (诸如线和边) 到高级语义概念 (诸如对象和部分) 的特征的层次结构的能力。因此, 预先训练的CNN可以被配置为有效且高效的特征生成器。例如, 作为特征生成器的、与简单的分类器 (诸如支持向量机 (support vector machine, SVM) 或逻辑回归) 耦合的预先训练的网络可以被配置为在各种任务上胜于手动调谐的特征。即使在与训练CNN所基于的原始任务非常不同的任务上, 也可以观察到这一点。因为如此, CNN可以广泛用作大型计算机视觉和机器学习应用管道 (诸如视频和图像分析) 中的特征计算模块。

[0030] 尽管与计算特征所根据的原始数据 (例如, 输入数据集合) 相比, 该特征在尺寸上往往较小, 但是对于较大的输入数据集合来说, 它们的存储仍然是禁止的。例如, YouTube-8M数据集合需要大约2万亿字节的内存。如果内存读取缓慢, 则基于这样的输入数据集合来训练神经网络的大管道就变得缓慢且昂贵。所描述的方法通过按照为压缩率和为原始训练目标而优化计算的特征的方式计算特征来解决这个问题。这产出了与初始的计算的特征集合一样强大的压缩的特征集合, 但只需要小部分的存储成本和内存。

[0031] 在一些实施方式中, 可以为靶向任务随时间演变的某些类型的应用预先计算和存储特征。一个示例是索引系统或内容分析系统, 其中要被索引的类别集合随时间而改变。因为类别不断地改变, 所以端到端连续地训练系统变得过分昂贵。另一示例是视频标签系统, 其中由于内存和计算限制, 分类模型的端到端训练可能是不可行的。这种系统通常被实施为管道, 其中首先计算特征并将其存储在数据存储库 (datastore) 中, 然后根据需要训练简单且高效的分类器。对于以互联网规模来操作的类似应用, 这种数据存储库可以增长到相当大的尺寸, 例如, 千万亿字节的范围。本文档中描述的用于生成压缩的特征的方法使得这种系统能够以小部分的成本操作, 而不会牺牲对于靶向任务的性能。

[0032] 来自CNN的特征可以通过移除上面几层并且使用剩余的最上层的激活来推导。这些特征往往是高维的, 当使用无损压缩方法存储时, 尤其是当大规模计算时, 需要大量的内存/存储空间。通常采用有损压缩方法 (诸如以降低的精度存储、事后 (post-hoc) 量化或降维方法 (例如, 主成分分析法 (principal component analysis, PCA))) 来降低存储成本。例如, 系统可以包含800万个视频的压缩的特征, 其中特征基于有损压缩以降低的存储成本被存储, 而特征的原始未压缩的尺寸可能由数百万亿字节组成, 这需要相当大的存储成本。这暗示了, 如果一个人要存储即使是很小部分的YouTube视频的特征, 则未压缩的表示也将需要数百千万亿字节。使用前述方法压缩这些特征会在特征中引入不想要的失真, 因为由压缩方法优化的目标没有考虑语义信息, 导致原始特征的辨别能力或信息能力的损失。

[0033] 因此, 本文档通过使用参考用于学习特征的靶向目标而对压缩的特征的精度和信息能力进行优化的方法, 解决了可压缩的特征的高效学习的问题。所描述的方法包括将特征层识别为瓶颈, 在该瓶颈处, 靶向目标鼓励信息流并与压缩率的目的背道而驰。所描述的方法引入了使得能够在特征集合的压缩率和信息量之间进行权衡的惩罚。所描述的方法的插件性质使其可以直接与任何靶向目标相结合。本文档证明, 基于这些方法而学习到的特性, 在保持其对靶向任务的信息量的同时, 其可压缩性与传统的压缩方法相比提高了几个数量级。

[0034] 图1示出了用于生成一个或多个特征集合的示例计算系统100。系统100通常包括示例的预先训练的神经网络110“CNN 110”，其处理输入数据集120以计算或生成特征集合。预先训练的CNN 110可以被表示为单个预先训练的神经网络130或多个预先训练的神经网络130的系统，其中系统的每个预先训练的神经网络130对应于相应的卷积神经网络或CNN。如下文更详细描述，CNN 110使用预先训练的神经网络系统130以生成用于生成一个或多个计算的中间表示135。

[0035] 如本文所描述的，中间表示135对应于CNN 110的瓶颈层，CNN 110可以用于降低计算的中间表示135的维数，而不损害计算的特征的预测能力。通常，计算的中间表示135可以由表示计算的特征的预测能力的信息量的度量(measure of informativeness)来表征。例如，示例信息量度量为0.9的第一特征集合比示例信息量度量为0.6的第二特征集合具有更高的预测能力。这里，相对于第一特征集合的预测能力，第二特征集合具有较低的预测能力。在一些情况下，特征集合的信息量由以下中的至少一个定义：特征集合中计算的中间表示135的精度度量；或者第一特征集合中的计算的中间表示135的维数。

[0036] 计算的中间表示135可以用于训练一个或多个分类器以执行某些分类函数，诸如用于识别图像中的对象的分类。更具体地，基于本文所描述的方法，系统100使用CNN 110根据计算的中间表示135来生成压缩的特征集合160。例如，CNN 110使用由压缩逻辑150指定的特定压缩方法以生成压缩的特征集合160。压缩逻辑150可以使用一种或多种压缩方法来压缩计算的中间表示135以生成压缩的特征集合160。在一些情况下，压缩的特征集合160在本文可替代地被称为特征的压缩集合160。

[0037] 系统100使用压缩的特征集合160来训练分类器170以生成训练后的分类器175。在一些实施方式中，分类器175被训练以生成分类输出180，诸如指示在图像或视频中被识别的项目的输出。以这种方式，系统100被配置为采用技术集合，包括使用CNN 110作为自动从输入数据集120中发现特征检测或分类所需的表示的特征生成器。

[0038] 图2A示出了使用图1的计算系统来生成压缩的特征集合160的示例过程200。过程200参考示例性监督的数据分类或回归问题来描述。在一些实施方式中，系统100对应于或包括在硬件电路上实施的一个或多个神经网络。在硬件电路上实施的神经网络可以包括预先训练的CNN，其被配置为特征生成器以执行与自动特征工程相关的一个或多个函数。如上所述，神经网络可以包括多个层，并且多个层中的至少一个层是(多个)神经网络的特征层。在一些实施方式中，神经网络是前馈特征检测器网络。

[0039] 系统100接收输入数据集120以用于生成从数据集推导出的特征(210)。例如，系统100可以通过使用神经网络的特征层之前的神经网络的一个或多个层计算第一特征集合的至少一部分来生成第一特征集合。在一些情况下，生成的特征可以是压缩的特征160。生成压缩的特征160可以包括将特征层识别为(多个)神经网络130的瓶颈层，以用于降低使用神经网络所计算的特征的维数。系统100使用中间表示135生成压缩的特征160。在一些实施方式中，系统100使用在CNN 110的瓶颈层处计算的中间表示135并基于损失函数来生成中间表示135。

[0040] 系统100可以被配置成相对于参数 $\theta$ 的集合来最小化损失函数L：

$$[0041] \quad \theta^* = \arg \min_{\theta} \sum_{x, y \in \mathcal{D}} L(\hat{y}, y), \quad \text{其中 } \hat{y} = f(x; \theta), \quad (1)$$

[0042] 在等式(1)中,  $x$ 是输入变量(例如,图像像素或特征),  $y$ 是目标变量(例如,分类标签或回归目标),  $\hat{y}$ 是预测,  $f$ 可以是人工神经网络(artificial neural network, ANN), 其中  $\theta$ 包括其滤波权重集合, 并且  $D$ 是训练数据集。

[0043] 系统100使用包括在CNN 110的神经网络130中的至少一个特征层, 根据输入数据集120来计算第一特征集合(例如, 计算的特征140)。第一特征集合:  $i$ ) 由信息量的度量表征; 并且  $ii$ ) 被计算以使得第一特征集合的尺寸可压缩为第二特征集合, 第二特征集合得到在尺寸上小于第一特征集合的尺寸的特征集合, 并且具有与第一特征集合的信息量的度量相同的信息量的度量(220)。

[0044] 通常, 计算特征对应于学习代表性特征的集合, 代表性特征可以用来取代输入数据集120中所包含的更大的信息集合。为了学习可用于取代  $x$  的代表性特征的集合, 系统100被配置为引入由变量  $z$  指示的中间表示135。在一些实施方式中, 中间表示135通过将  $f$  划分为两部分  $f_1$  和  $f_2$  而生成。那么预测  $\hat{y}$  由下式给出:

$$[0045] \quad \hat{y} = f(x; \theta) = f_2(z; \theta_2), \quad \text{其中 } z = f_1(x; \theta_1), \quad (2)$$

[0046] 如上所述, 通常把像  $z$  这样的中间表示称为瓶颈(bottleneck)。在许多情况下, 例如在自动编码器的上下文中, 瓶颈用于在不损害预测能力的情况下降低维数, 即迫使  $z$  具有比  $x$  更小的维度的数量。类似地, 可以使用CNN 110的瓶颈层来降低计算的特征140的维数, 而不削弱根据计算的特征140而生成的压缩的特征集合160的预测能力。因为维度的数量是超参数(例如, 架构选择), 所以不需要对损失函数  $L$  进行改变, 并且简单地训练示例神经网络模型以在给定约束下最小化损失。然而, 降维只是数据压缩的粗略近似。

[0047] 系统100使用压缩第一特征集合以生成第二特征集合的压缩方法, 根据第一特征集合来生成第二特征集合(例如, 特征集合160)(230)。因此, 第二特征集合是表示第一特征集合的压缩版本的压缩的特征集合160。在一些实施方式中, 输入数据集120包括图像数据, 并且生成第二特征集合包括训练CNN 110的神经网络130以学习指示数据集的图像数据的特性的特征的层次结构。特征的层次结构可以从包括图像中描述的项目的线和边的低级图像特征到与数据集的对象相关的高级语义特征。

[0048] 系统100响应于将压缩逻辑150的压缩方法应用于根据中间表示135而计算的第一特征集合来生成第二特征集合。通常, 数据压缩通过这样的方式来工作: 将每个可能的数据点映射到符号(通常为比特)的可变长度, 存储或传输它们, 并且在接收器侧反转该映射。

[0049] 更具体地, 存储离散值的数据集合  $Z$  所需的最佳比特数量由香农熵给出:

$$[0050] \quad H = - \sum_{z \in Z} \log_2 p(z), \quad (3)$$

[0051] 其中  $p$  是数据点的概率分布, 其需要对数据的发送者和数据的接收者都可用。熵编码(entropy coding)技术(诸如算术编码或霍夫曼编码)使用该概率分布以实施映射。

[0052] 在一些实施方式中, 生成中间表示135包括将熵项应用于损失函数  $L$  以获得增强的损失函数。中间表示135可以通过使用增强的损失函数计算在神经网络的瓶颈层处的特征而生成。在一些情况下, 使用中间表示135计算的特征具有受到熵项影响的压缩率的度量, 该熵项被应用以获得增强的损失函数。例如, 瓶颈表示的压缩率可以针对预测或回归误差进行权衡, 直到确定该误差足以用于特定应用, 从而实现更大的存储节省。这通常被称为有

损压缩 (lossy compression)。为了能够以不同的权衡为目标,系统100被配置为用熵项来增强损失函数:

$$[0053] \quad \theta^*, \phi^* = \arg \min_{\theta, \phi} \sum_{x, y \in \mathcal{D}} L(\hat{y}, y) + \lambda \cdot \underbrace{-\log_2 p(\hat{z}; \phi)}_{\text{比特率 } R}, \quad (4)$$

[0054] 其中,  $\hat{z} = \lfloor f_1(x; \theta_1) \rfloor$ ,  $\hat{y} = f_2(\hat{z}; \theta_2)$ , 并且  $p$  是针对  $\hat{z}$  的具有参数  $\phi$  的概率模型, 该参数  $\phi$  与  $\theta$  联合训练。 $\lfloor \cdot \rfloor$  在这里指示  $f_1$  的输出被四舍五入到最接近的整数, 并且与损失函数的量化元素相关联。这种量化是有用的, 因为压缩发生在具有数量可计的可能状态的离散空间中。这是在压缩性能和预测性能之间实现权衡的方式。例如, 当  $\hat{z}$  中的可能状态的数量降低时 (例如, 通过按比例缩小  $f_1$  的输出), 比特率  $R$  可以以预测性能为代价而降低, 因为量化随后在  $\hat{z}$  中引入了更大的相对误差。相反, 以压缩率为代价, 可以通过增加可能状态的数量来提高预测性能。超参数对应于权重变量  $\lambda$ , 权重变量  $\lambda$  用于控制率项 (rate term) 的权重, 从而控制期望的权衡。这种类型的瓶颈可以被称为熵 (entropy) 或信息瓶颈 (information bottleneck)。

[0055] 在一些实施方式中, 损失函数可以包括量化元素, 并且将熵项应用于损失函数以获得增强的损失函数可以包括在神经网络 (诸如用于获得概率模型的第二神经网络) 的示例训练阶段期间用加性均匀噪声元素替代损失函数的量化元素。例如, 用下降方法直接最小化上述损失函数可能是不可行的, 因为量化会导致几乎处处为零的梯度。相反, 系统100可以在训练期间用加性均匀噪声替代量化。

[0056] 训练期间均匀噪声的添加允许系统对可能由量化导致的误差进行模拟, 而不实际进行或执行量化 (例如, 诸如梯度为零从而不能用于训练的量化)。在一些情况下, 这种替代可以用于对率进行建模。下面将更详细地描述与这种加性噪声方法相关的技术。对于失真, 系统可以例如通过四舍五入进行离散化, 并且直接由恒等 (identity) 替代梯度。

[0057] 此外, 作为使用分段线性密度模型的替代, 可以改为使用更精细的密度模型。例如, 分段线性密度模型本身可能是受限制的, 因此可能不能很好地用于对密度进行建模。可以改为使用更精细的密度模型来获得与密度建模相关的更好的结果。

[0058] 在一些实施方式中, 等式4的熵惩罚的权重  $\lambda$  可以影响概率模型  $p$  的参数  $\Phi$  的梯度更新的幅度。较小的  $\lambda$  值可以降低  $\Phi$  的有效学习率, 使模型学习更慢。这可能导致观察到的分布与模型之间脱节。Adam 优化器可以用于计算由平方梯度的滑动平均值 (running average) 的平方根所归一化的更新。这具有期望的性质, 即恒定的损失缩放不会影响更新的幅度。因此, 对于等式4中的组合的损失, 由于熵惩罚,  $\lambda$  只影响梯度的相对权重, 而不改变  $\Phi$  的有效学习率。

[0059] 量化/四舍五入的变量的概率质量函数通过针对映射到相同整数的所有值对概率密度进行积分来给出:

$$[0060] \quad p(\hat{z}) = \int_{\hat{z}-\frac{1}{2}}^{\hat{z}+\frac{1}{2}} p(z) dz = c(\hat{z} + \frac{1}{2}) - c(\hat{z} - \frac{1}{2}), \quad (5)$$

[0061] 其中  $\hat{z}$  是任何整数值, 并且  $c$  是  $p$  的累积。类似地, 遭受加性标准均匀噪声的变量的

概率密度由下式给出：

$$[0062] \quad p(\tilde{z}) = \int_{\tilde{z}-\frac{1}{2}}^{\tilde{z}+\frac{1}{2}} p(z) dz = c(\tilde{z} + \frac{1}{2}) - c(\tilde{z} - \frac{1}{2}), \quad (6)$$

[0063] 其中  $\tilde{z}$  是任何连续的值。注意等式 (5) 和 (6) 两个表达式是相同的：噪声变量的密度在非整数值处对量化变量的概率质量函数进行插值。这一事实被用来为训练创建有差别的代理损失。

[0064] 本着对流进行归一化，所使用的密度模型被直接定义为上面的累积量的差。为了创建有效的密度模型， $c$  可以是满足下列性质的任何函数  $c: \mathbb{R} \rightarrow [0, 1]$ ：

$$[0065] \quad c(-\infty) = 0; \quad c(\infty) = 1; \quad \frac{\partial c(x)}{\partial x} \geq 0。 \quad (7)$$

[0066] 这可以通过使用神经网络 (NN) 实施密度模型来实现，其中最后的激活函数是逻辑函数，确保满足前两个要求。表达式 (7) 的第三个单调性要求通过只允许单调激活函数并确保网络中所有线性权重为正来保证。

[0067] 对于本文档中描述的一个或多个示例，熵模型可以包括具有零个或多个隐藏层的神经网络。在一些示例中，熵模型可以包括具有至少一个隐藏层的神经网络，每个隐藏层具有至少一个隐藏单元。层的数量可以对应于密度模型的整体性能。例如，增加熵模型中所包括的层的量将转化为产出更好性能结果的更强大的密度模型。

[0068] 对于  $\hat{\mathbf{z}}$  中的每个向量元素  $\hat{\mathbf{z}}_i$ ，可以使用单独的模型，产出完全因子化的概率模型：

$$[0069] \quad p(\hat{\mathbf{z}}) = \prod_i p(\hat{\mathbf{z}}_i)。 \quad (8)$$

[0070] 在具有空间配置的瓶颈的情况下，系统100被配置为使得相同通道内的空间元素具有相同的分布（例如，密度模型的参数在空间上被共享）。如上所述，系统100使用压缩的特征集合160来训练分类器170以生成训练后的分类器175。因此，压缩的特征集合160是用于训练示例神经网络分类器170的数据，以基于压缩的特征集合160中的特征的特性来检测输入数据集120中的数据对象的特性。在一些实施方式中，通过经由神经网络分类器的一个或多个层处理压缩的特征集合160来训练神经网络分类器170，以生成训练后的分类器175。

[0071] 如下所述，可以针对代表普通的现成的特征生成方法的一个或多个示例分类模型来评估本文档中描述的改进方法。来自一些示例评估的数据集合证明，上述改进方法可以产出计算的特征140，计算的特征140比使用普通的现成方法所获得的特征明显更具可压缩性，并且不会损失与从普通的方法得到的预测能力相关的表达性 (expressiveness) 或信息量。在一些实施方式中，示例评估涉及将紧随非线性之后的倒数第二层的激活视为特征层的标准实践。例如，系统100可以被配置为将该特征层视为瓶颈  $\hat{\mathbf{z}}$ ，并对其应用率惩罚。

[0072] 图2B示出了与用于训练神经网络以生成压缩的特征集合 (260) 和用于评估或测试压缩的特征集合 (270) 的机器学习方法相关联的示例框图250。

[0073] 在图2B的实施方式中， $f_1$  根据输入图像  $x$  来产生中间表示  $z$ ，其中输入图像可以对应于上述输入数据集120中所包括的图像。在训练期间，添加均匀噪声262以模拟量化，同

时允许基于梯度的优化。熵模型264用于对 $\hat{z}$ 的分布进行建模并施加压缩损失 $R(\hat{z})$ 。然而，在测试期间，使用四舍五入(Q)来量化中间表示 $z$ 以产出 $\hat{z}$ ，然后使用算术编码(arithmetic encoding, AE)来产生用于存储的可变长度比特流(272)。可以使用算术解码(arithmetic decoding, AD)来对该可变长度比特流进行解码以产出 $\hat{z}$ ，并且确定与任务输出相对应的最终预测。本文描述了与图2B的实施方式相关联的其他细节，并且至少在以上对图1和图2A的描述中可以参考这些细节。

[0074] 图3A和图3B各自示出了关于与根据相应的标记的输入数据集合而生成的压缩的特征集合相关联的分类误差的图形数据。图3A示出了图形数据310，图形数据310指示针对CIFAR-10图像数据集合的相对压缩比而观察到的分类误差。类似地，图3B示出了图形数据320，图形数据320指示针对CIFAR-100图像数据集合的相对压缩比而观察到的分类误差。CIFAR-10(图3A)和CIFAR-100(图3B)各自分别是包含10个和100个图像类的图像分类数据集合。在一些实施中，CIFAR-10和CIFAR-100各自包含50000个训练和10000个测试图像。

[0075] 如图3A和图3B所述，本文档的改进方法(例如，如上所述)可以产生信息丰富但高度可压缩的特征表示。图形数据310、320将解压缩的特征的分类误差可视化为相对于CIFAR-10(图3A)和CIFAR-100(图3A)上的无损压缩的相对压缩比的函数。在一些实施方式中，这些解压缩的特征对应于使用系统100的CNN 110生成的压缩的特征集合160的解压缩版本。

[0076] 对于CIFAR-10，改进方法可以产生尺寸为无损压缩时的1%的表示，同时具有相同的准确度。类似地，对于CIFAR-100，由改进方法产生的表示在无损压缩的尺寸的10%处保持准确度，同时展现出类似的分类误差。

[0077] 如图3A和图3B所示，评估的结果数据集合可以以示出分类误差相对于相对的压缩的示例曲线图的形式而呈现。相对的压缩被测量为由下面描述的无损压缩基线所实现的压缩的尺寸的一小部分。

[0078] 考虑基线压缩方法，可以通过将其与下面讨论的几种标准方法进行比较并使用可以包括多个不同图像和图像类的输入数据的样本集合，来评估改进方法。对于这些方法，输入数据集合中每个图像的特征表示以浮点32精度计算并被独立地压缩。这些方法的性能可以被报告为以特定压缩率实现的分类误差，该特定压缩率被报告为无损压缩的尺寸的一小部分(例如，相对压缩比)。

[0079] 在一些实施方式中，对于无损压缩，使用Python中与gzip兼容的zlib压缩库来压缩表示。例如，每个表示首先被转换为字节数组，然后使用zlib以最高压缩级别9进行压缩。对于16比特gzip，表示首先被运算为16比特浮点表示，然后被无损压缩。对于量化，表示被缩放到单位范围，然后量化到等长间隔。性能可以被表示为集合 $\{2^{16}, 2^8, 2^4, 2^2\}$ 中的量化二进制数的数量的函数。使用gzip再次无损压缩这些量化值，以获得额外的压缩增益。如果使用的量化二进制数少于256个，则在使用gzip之前，数据会以字节形式(未封装)本地存储。

[0080] 对于诸如主成分分析(PCA)的降维方法，基于相对于训练集合而计算的表示来估计完全协方差矩阵，并且计算其特征分解(eigen-decompositio)以产出主成分。性能被表示为在集合 $\{1, 2, 4, 8, 16, 32, 64\}$ 中增加的所使用成分的数量函数的函数。PCA基础可以被视为压缩模型的部分，并且被排除在压缩成本的计算之外。在一些实施方式中，系数以32比特浮

点精度被存储,并且原始的比特数量可以被报告为尺寸,因为使用zlib的压缩可能会始终增加文件尺寸。

[0081] 在一些实施方式中,基本分类模型是两个CIFAR数据集合的20层ResnetV2模型。可以使用SGD进行128000次迭代,并且使用初始学习率为0.005的余弦衰减学习率时间表来训练神经网络。可以使用左右翻转、并用4像素对所有边进行零填充、然后进行 $32 \times 32$ 裁剪的标准数据增强。在一些情况下,当在使用128的批处理尺寸的单个图形处理器单元(Graphics Processor Unit,GPU)上训练模型时,可以使用0.0001的权重衰减。

[0082] 关于学习的特征表示的压缩率,图3A和图3B各自指示改进方法始终产生作为比无损压缩时更具可压缩性的数量级的表示,同时保持或很大程度上保持未压缩的学习的表示的辨别能力和预测能力。在一些实施方式中,解压缩的特征的分类误差被可视化为相对于无损压缩的相对压缩比的函数。

[0083] 如上所述,对于CIFAR-10,改进方法可以产生无损压缩尺寸的1%的表示,同时与精度相匹配。这很可能是由于只有10个类的事实,10个类理想情况下只需要 $\log_2 10$ 个比特。对于CIFAR-100,改进方法可以产生压缩为无损压缩尺寸的10%的表示,同时保持相同的精度。对于CIFAR-100,相对于CIFAR-10,在所实现的压缩上数量级降低,在类别的数量上数量级增加(例如,从10到100)。

[0084] 对于CIFAR-10和CIFAR-100数据集合,16比特gzip始终保持性能,表明16比特精度对于这些表示来说是足够准确的。随着量化二进制数的数量的减少,量化可能会失去性能。在一些实施方式中,PCA比其他基线表现更好,然而因为使用较少的成分,其性能可能会削弱。

[0085] 在一些实施方式中,所描述的方法可以具有正则化效应。除了纯有损压缩之外,在一些情况下,所提出的方法充当激活正则化器,允许基于验证集合的分类结果高于当使用原始网络时的分类结果,同时展现出更高的训练误差(例如,见下面的表1)。这种正则化效应可以被描述为具有甜点(sweet spot),其可以为要解决的问题的复杂性提供一些洞察。与正常的正则化方法不同,所描述的方法在编码器网络和分类器之间传递的信息之间进行权衡。因此,可以明确地测量特定分类任务所需的信息量。在一些实施方式中,CIFAR-100需要较少的压缩以获得最佳结果,而CIFAR-10需要大约一半的信息以便获得最佳结果。这表明,为解决这两个问题而设计的网络可能会比CIFAR-10情况下所需的网络稍大。

[0086] 图4示出了指示与根据特定类型的标记的数据集合而生成的压缩的特征集合相关联的分类误差的示例图形数据410。例如,可以使用具有1000个图像类的ImageNet分类数据集来评估(多个)改进方法。在ImageNet上,改进方法保留了初始特征集合的准确度,同时将存储成本降低到无损压缩的文件尺寸的大约12%。

[0087] 该示例评估可以包括基于大约120万个训练图像来训练模型,并且基于50000个验证图像的集合来报告结果。在一些实施方式中,50层ResnetV2模型可以用作评估的基础模型。可以使用SGD进行300000次迭代,并且使用初始学习率为0.001的余弦衰减学习率时间表来训练神经网络。当基于 $299 \times 299$ 像素的图像裁剪进行训练时,可以使用标准的数据增强。在一些实施方式中,当在8个GPU上训练每个模型时,可以使用0.0001的权重衰减,每GPU具有32的批处理尺寸,从而产生256的组合批处理尺寸和同步更新。在一些情况下,评估包括报告基于 $299 \times 299$ 像素中心裁剪而计算的前1(top-1)分类误差。

[0088] 关于学习的特征表示的压缩率,图4的图形数据指示了类似于如上参考图3A和图3B描述的基于CIFAR-10/100数据集合而观察到的趋势。与其他基线方法相比,改进方法产生了高度可压缩的表示,并且能够保持准确度,同时将存储成本降低到无损压缩的文件尺寸的12%。在一些实施方式中,16比特精度的无损存储会导致在误差上增加0.14%。与CIFAR-10/100数据集合类似,如下表1所示,尽管验证性能更好,但相比于基线仍可能出现基于训练集合误差更高的正则化效应。

[0089] 图5A示出了指示与使用两层神经网络生成的压缩的特征集合相关联的分类误差的图形数据510。图5B示出了指示与使用三层神经网络生成的压缩的特征集合相关联的分类误差的图形数据520。图5A和图5B各自的图形数据510、520是基于使用Youtube-8M(“YT-8M”)数据集合的改进方法的示例评估的。YT-8M数据集合是包括超过600万个视频和3862个图像类的大型公开可用视频分类数据集合。在YT-8M数据集合上,改进方法保持初始的压缩的特征集合的压缩版本的准确度,同时将存储成本降低到无损压缩的文件尺寸的5%到10%之间。可以使用2层网络(图5A)和3层网络(图5B)来执行所描述的方法相比于其他常规方法的评估和比较。

[0090] 在一些实施方式中,示例评估包括首先使用均值池化(mean pooling)将视频序列特征集合聚合成固定尺寸的向量。可以使用作为由两个或三个全连接层组成的神经网络的基本模型。神经网络可以包括其隐藏层上的ReLU激活和批处理归一化(“BatchNorm”),以及输出层上的无BatchNorm的逻辑激活。可以使用10<sup>-6</sup>的权重衰减,并且可以用100的批处理尺寸、最小化交叉熵损失、以及使用TensorFlow的Adam优化器进行300000个步骤在一个CPU上训练每个模型。可以从选择{0.04,0.02,0.002,0.001,0.0005}中扫视初始学习率,并且学习率可以是每1000个步骤乘以0.94。对于每个模型架构,根据留存的(held-out)验证集合来使用优选的学习率。压缩可以应用于就在输出层之前的最后的隐藏激活。

[0091] 关于学习的特征表示的压缩率,改进方法可以用于显著降低存储语义视频信息的压缩的特征的存储要求,同时还保持(或略微提高)泛化准确度。准确度量度是基于YT-8M数据集合的“验证”分区而测量的。这可能潜在地对在线视频系统有很大影响。通过推断,假设具有10亿个视频的假想的在线视频系统试图使用视听内容来对视频注解以支持视频搜索端点。存储均值池化的视听特征(例如,与YT-8M中可用的特征相同)将需要原始存储形式的4.6万亿字节的存储。然而,如果使用本文中描述的改进方法来训练注解系统,那么存储需求将下降到小于1500亿字节,同时潜在地提高注解新视频的泛化性能。

	训练误差		验证误差		验证集合尺寸		
	无损	方法	无损	方法	无损	方法	原始
[0092] ImageNet	17.04	17.35	25.91	25.89	6.95GB	0.85GB	38.15GB
CIFAR-10	0.14	0.29	8.73	8.45	41.53MB	2.78MB	156.25MB
CIFAR-100	0.25	1.54	33.39	33.03	69.14MB	9.28MB	156.25MB
YouTube-8M	19.56	19.75	19.76	19.49	5.30GB	0.27GB	17.80GB

[0093] 经由上表1所示,改进方法除了具有正则化效应之外,还可以显著降低存储成本。评估数据集合的总压缩的尺寸以及训练后的模型的最终训练误差和验证误差可以与在没

有率惩罚情况下训练的基线模型进行比较。对于每个数据集，选择误差低于基线的最低率模型。对于YT-8M数据集，所报告的尺寸属于视频级特征。对于使用本文档的改进方法训练的模型，训练误差和验证误差之间的差距始终较小。这表明熵惩罚具有正则化效应，而使用改进方法训练的模型显著降低了压缩的特征集合的总尺寸。

[0094] 关于压缩的联合优化，普通或常规的方法提出通过相对于辅助任务的瓶颈添加额外的推断分支来扩展自动编码压缩网络。然而，本文档的改进方法不使用任何自动编码惩罚，而是直接针对熵以及任务特定目标进行优化。因此，所描述的方法不需要域(domain)适于自动编码。此外，改进方法可以被插入到任何架构中作为额外的惩罚，而无需架构改变或结构约束。

[0095] 各种降维方法都在本公开的范围内。尽管不尝试信息理论的压缩，但是有各种可以被应用以最小化预先计算的CNN特征所需的空间的有损降维方法。例如，可以使用PCA、线性判别分析(Linear Discriminant Analysis, LDA)或独立成分分析(Independent Component Analysis, ICA)。在一些实施方式中，也可以使用乘积量化。然而，与本文档中描述的改进方法相比，这些方法都没有考虑到需要被最小化的真正损失。在一些情况下，这些方法可以依靠替代损失(通常是 $L_2$ )。在神经网络的上下文中，也可以使用自动编码器来降低损失。自动编码器可以通过改变隐藏表示所需的维度的数量来实现压缩，并且自动编码器是图像压缩的典型场景。

[0096] 参考变化的信息瓶颈，所描述的方法是更一般的信息瓶颈框架的特定实例。在一些情况下，互信息可以对应于信息量的参数化独立度量，通常可以使用任务相关的度量，并且如果提前知道靶向任务，则该任务相关的度量可能更适合。大多数分类模型通常被训练来优化交叉熵，并且相同的过程可以在本文档中用作信息量的度量。

[0097] 如上所述，用于生成多个压缩的特征集合的一些方法可以包括使用来自现成压缩算法的压缩的表示。例如，压缩的表示可以直接用于训练机器学习算法。这种方法可以是有吸引力的，因为压缩的表示的很低的内存和计算要求能够实现高效的实时处理，同时还避免了解码开销。本文描述的改进方法可以不同于这些现有的方法，因为所描述的方法与靶向任务一起对可压缩的表示进行联合优化。相比之下，现有的方法使用现成的压缩方法，这些现有的压缩方法为重建目标而进行优化并且不能适应靶向任务。在一些情况下，为像素分类配置的线性SVM被用在压缩的高光谱数据上。在其他实施方式中，可以使用自适应概率方法在压缩的域上执行高光谱像素分类。此外，一些方法包括将离散余弦变换(“DCT(discrete cosine transform)”)压缩的图像数据馈送到网络中，以加速应用在图像上的机器学习算法。

[0098] 本说明书中描述的主题和操作的实施例可以以数字电子电路来实施，或者以计算机软件、固件或硬件(包括本说明书中公开的结构及其结构等价物)来实施，或者以它们中的一个或多个的组合来实施。本说明书中描述的主题的实施例可以被实施为一个或多个计算机程序，即计算机程序指令的一个或多个模块，该计算机程序指令被编码在计算机存储介质上，用于由数据处理装置执行或控制数据处理装置的操作。

[0099] 可替代地或附加地，程序指令可以被编码在人工生成的传播信号(例如，机器生成的电、光或电磁信号)上，人工生成的传播信号被生成以对信息进行编码，以用于传输到合适的接收器装置以供数据处理装置执行。计算机存储介质可以是或可以被包括在计算机可

读存储设备、计算机可读存储基底、随机或串行存取存储器阵列或设备、或它们中的一个或多个的组合中。此外，尽管计算机存储介质不是传播信号，但是计算机存储介质可以是编码在人工生成的传播信号中的计算机程序指令的源或目的地。计算机存储介质也可以是或也可以被包括在一个或多个单独的物理组件或介质（例如，多个CD、磁盘或其他存储设备）中。

[0100] 本说明书中描述的操作可以被实施为由数据处理装置对存储在一个或多个计算机可读存储设备上或从其他源接收的数据执行的操作。

[0101] 术语“数据处理装置”涵盖用于处理数据的所有种类的装置、设备和机器，包括例如可编程处理器、计算机、片上系统、或者前述的多个或组合。该装置可以包括专用逻辑电路，例如FPGA（现场可编程门阵列）或ASIC（专用集成电路）。除了硬件之外，该装置还可以包括为所讨论的计算机程序创建执行环境的代码，例如，构成处理器固件、协议栈、数据库管理系统、操作系统、跨平台运行时环境、虚拟机或它们中的一个或多个的组合的代码。该装置和执行环境可以实现各种不同的计算模型基础设施，诸如网络服务、分布式计算和网格计算基础设施。

[0102] 计算机程序（也称为程序、软件、软件应用、脚本或代码）可以以任何形式的编程语言（包括编译语言或解释语言、声明性语言或过程性语言）来编写，并且其可以以任何形式（包括作为独立的程序或作为模块、组件、子例程、对象或其他适合在计算环境中使用的单元）来部署。计算机程序可以但不必对应于文件系统中的文件。程序可以存储在保留其他程序或数据的文件的部分（例如，在标记语言文档中存储的一个或多个脚本）中，存储在专用于所讨论的程序的单个文件中，或者存储在多个协调文件（例如，存储一个或多个模块、子程序或部分代码的文件）中。计算机程序可以被部署为在一个计算机上或者在位于一个站点或分布在多个站点并通过通信网络互连的多个计算机上执行。

[0103] 本说明书中描述的过程和逻辑流程可以由执行一个或多个计算机程序的一个或多个可编程处理器来执行，以通过对输入数据进行操作并生成输出来执行动作。该过程和逻辑流程也可以由专用逻辑电路来执行，并且装置也可以被实施为专用逻辑电路，例如，FPGA（现场可编程门阵列）或ASIC（专用集成电路）。

[0104] 例如，适合于执行计算机程序的处理器包括通用和专用微处理器以及任何类型的数字计算机的任何一个或多个处理器。通常，处理器将从只读存储器或随机存取存储器或者两者接收指令和数据。计算机的基本元件是用于根据指令执行动作的处理器和用于存储指令和数据的一个或多个存储器设备。

[0105] 通常，计算机还将包括用于存储数据的一个或多个大容量存储设备（例如磁盘、磁光盘或光盘）或者可操作地耦合到一个或多个大容量存储设备，以从其接收数据，或向其传送数据，或两者兼有。然而，计算机不需要这样的设备。此外，计算机可以嵌入到另一设备中，例如移动电话、个人数字助理（PDA）、移动音频或视频播放器、游戏机、全球定位系统（GPS）接收器或便携式存储设备（例如通用串行总线（USB）闪存驱动器），仅举几个示例。

[0106] 适合于存储计算机程序指令和数据的设备包括所有形式的非易失性存储器、介质和存储器设备，包括例如，半导体存储器设备（例如，EPROM、EEPROM和闪存设备）；磁盘（例如，内部硬盘或可移动磁盘）；磁光盘；以及CD-ROM和DVD-ROM盘。处理器和存储器可以由专用逻辑电路来补充或结合于专用逻辑电路。

[0107] 为了提供与用户的交互，本说明书中描述的主题的实施例可以在计算机上实施，

该计算机具有用于向用户显示信息的显示设备(例如CRT(cathode ray tube,阴极射线管)或LCD(liquid crystal display,液晶显示器)监控器)以及用户可以通过其向计算机提供输入的键盘和定点设备(例如,鼠标或轨迹球)。

[0108] 也可以使用其他类型的设备来提供与用户的交互;例如,提供给用户的反馈可以是任何形式的感觉反馈,例如,视觉反馈、听觉反馈或触觉反馈;并且来自用户的输入可以以任何形式被接收,包括声音、语音或触觉输入。此外,计算机可以通过向用户使用的设备发送文档和从用户使用的设备接收文档来与用户交互;例如,通过响应于从网络浏览器接收到的请求,向用户的用户设备上的网络浏览器发送网页。

[0109] 本说明书中描述的主题的实施例可以在计算系统中实施,该计算系统包括后端组件(例如,作为数据服务器),或包括中间件组件(例如,应用服务器),或包括前端组件(例如,具有图形用户界面或网络浏览器的用户计算机,用户可以通过该图形用户界面或网络浏览器与本说明书中描述的主题的实施方式进行交互),或者包括一个或多个这样的后端、中间件或前端组件的任何组合。系统的组件可以通过任何形式或介质的数字数据通信(例如,通信网络)来互连。通信网络的示例包括局域网(“LAN”)和广域网(“WAN”)、互连网络(例如,互联网)和对等网络(例如,adhoc对等网络)。

[0110] 计算系统可以包括用户和服务器。用户和服务器一般彼此远离,并且通常通过通信网络进行交互。用户和服务器的关系是凭借运行在相应计算机上并且彼此之间具有用户-服务器关系的计算机程序而产生的。在一些实施例中,服务器向用户设备传输数据(例如,HTML页面)(例如,为了向与用户设备交互的用户显示数据和从该用户接收用户输入的目的)。可以在服务器处从用户设备接收在用户设备处生成的数据(例如,用户交互的结果)。

[0111] 尽管本说明书包含许多特定的实施方式细节,但这些不应被解释为对任何发明或可能要求保护的范围的限制,而是对特定发明的特定实施例的特征的描述。本说明书中在单独的实施例的上下文中描述的某些特征也可以在单个实施例的组合中实施。

[0112] 相反,在单个实施例的上下文中描述的各种特征也可以在多个实施例中单独地或以任何合适的子组合来实施。此外,尽管特征可以在上面被描述为以某些组合起作用,甚至最初也是这样要求保护的,但是在一些情况下,来自要求保护的组合的一个或多个特征可以被从该组合中删除,并且要求保护的组合可以针对于子组合或子组合的变体。

[0113] 类似地,尽管在附图中以特定次序描述了操作,但这不应该被理解为要求以所示的特定次序或顺序次序执行这些操作,或者执行所有示出的操作,以实现期望的结果。在某些情况下,多任务和并行处理可能是有利的。此外,上述实施例中的各种系统组件的分离不应被理解为在所有实施例中都需要这样的分离,并且应当理解,所描述的程序组件和系统一般可以一起集成在单个软件产品中或者封装到多个软件产品中。

[0114] 因此,已经描述了主题的特定实施例。其他实施例在权利要求的范围内。在一些情况下,权利要求中列举的动作可以以不同的次序执行,并且仍然实现期望的结果。此外,附图中描述的过程不一定需要所示的特定次序或顺序次序来实现期望的结果。在某些实施方式中,多任务和并行处理可能是有利的。

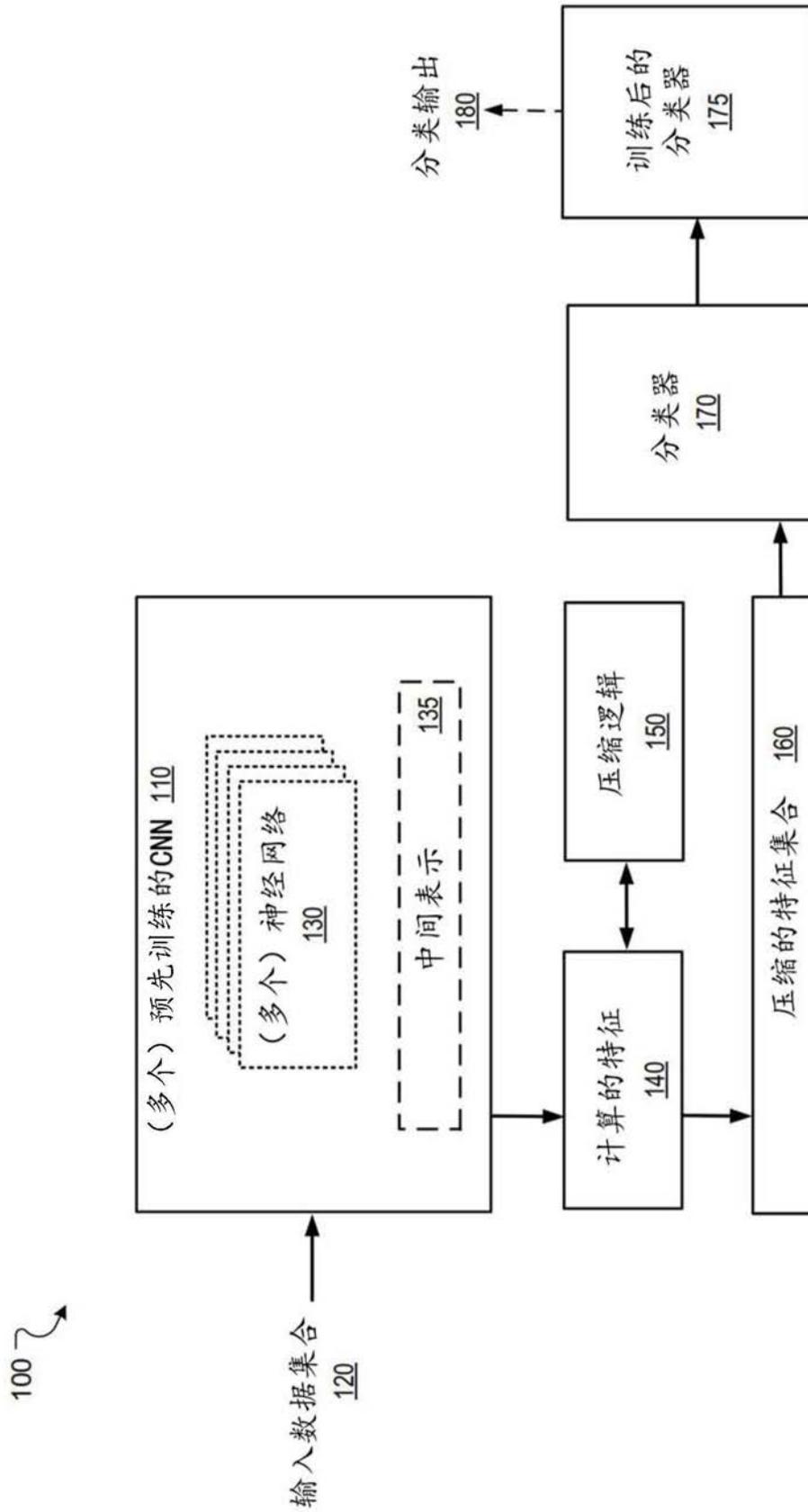


图1

200 ↘

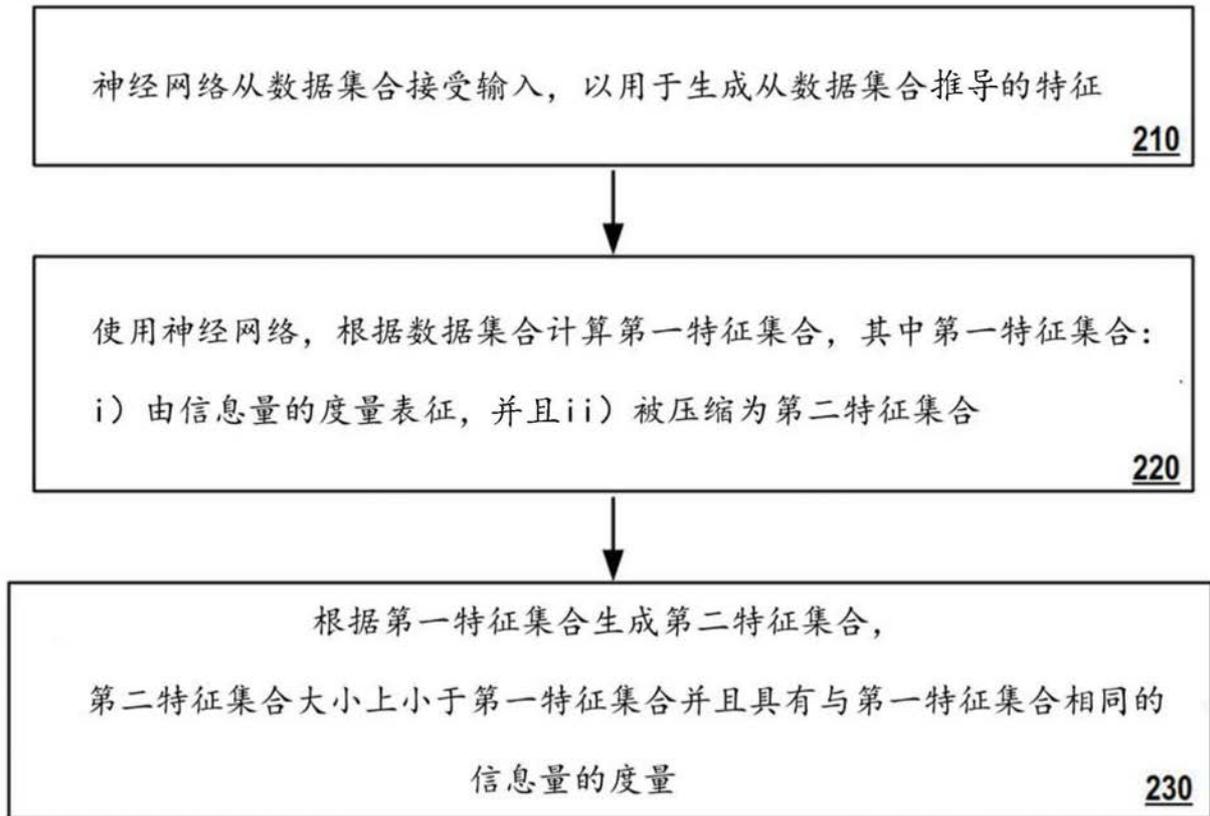


图2A

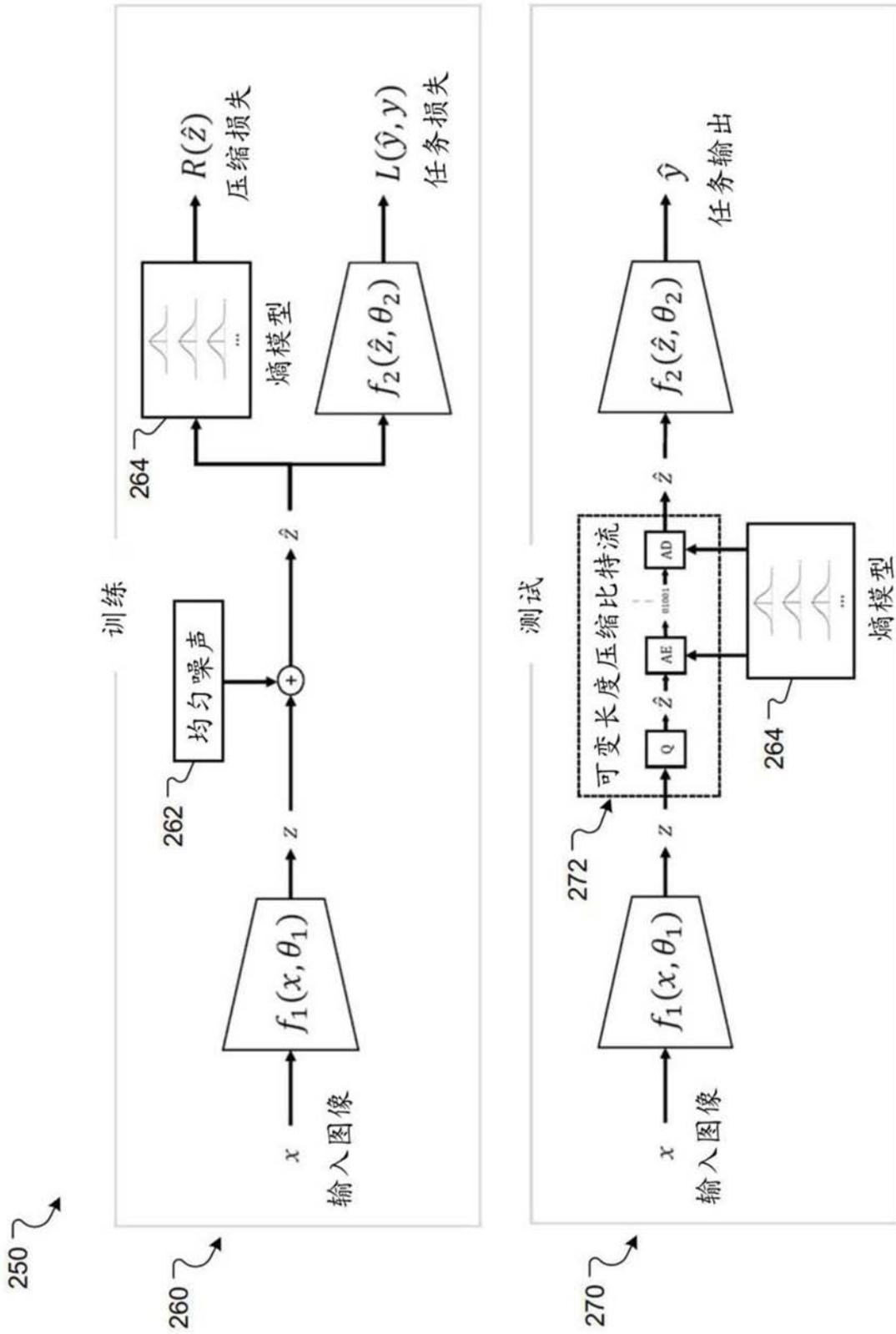
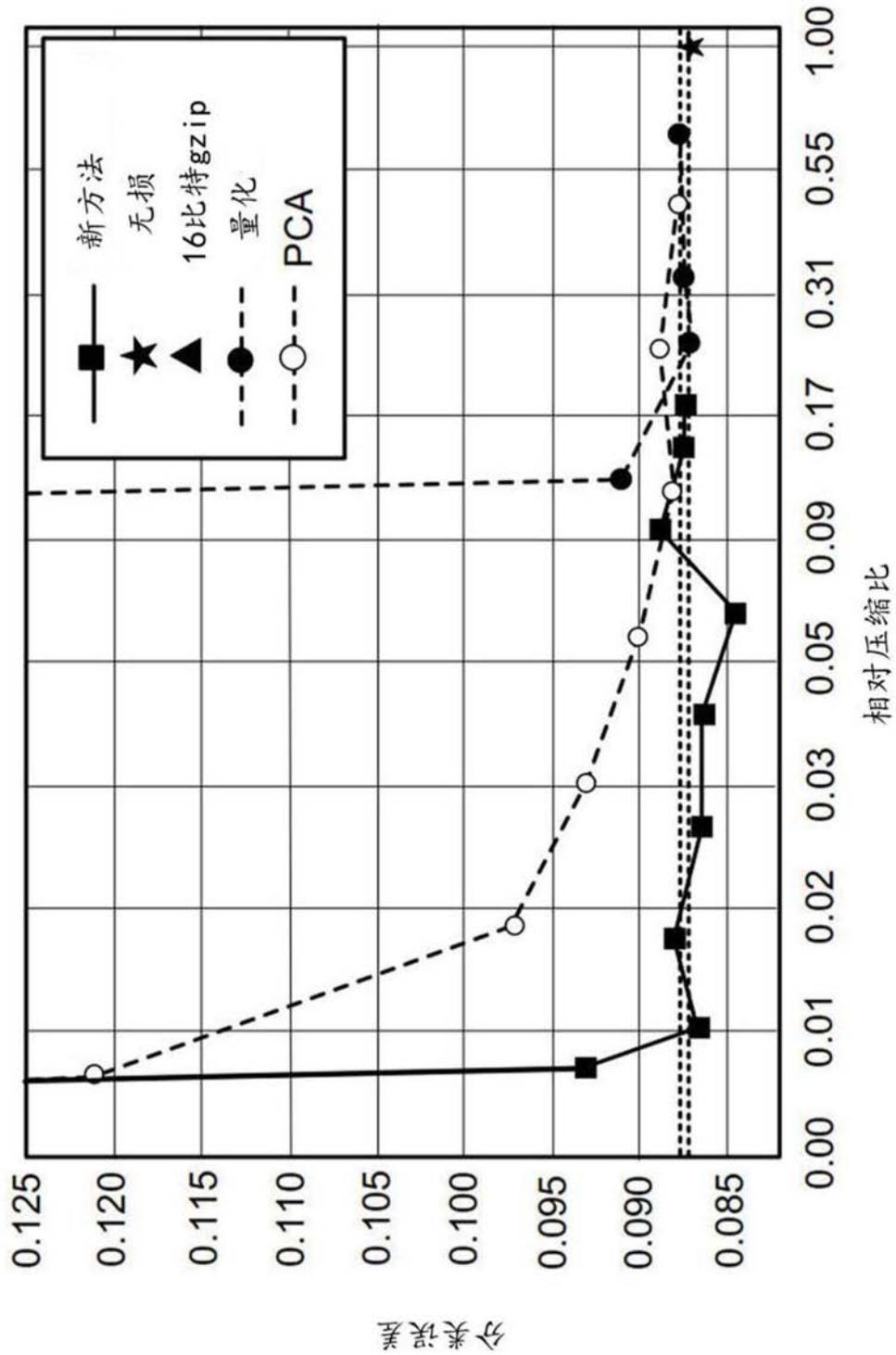


图2B



(a) CIFAR-10

310 ↷

图3A

320 ↗

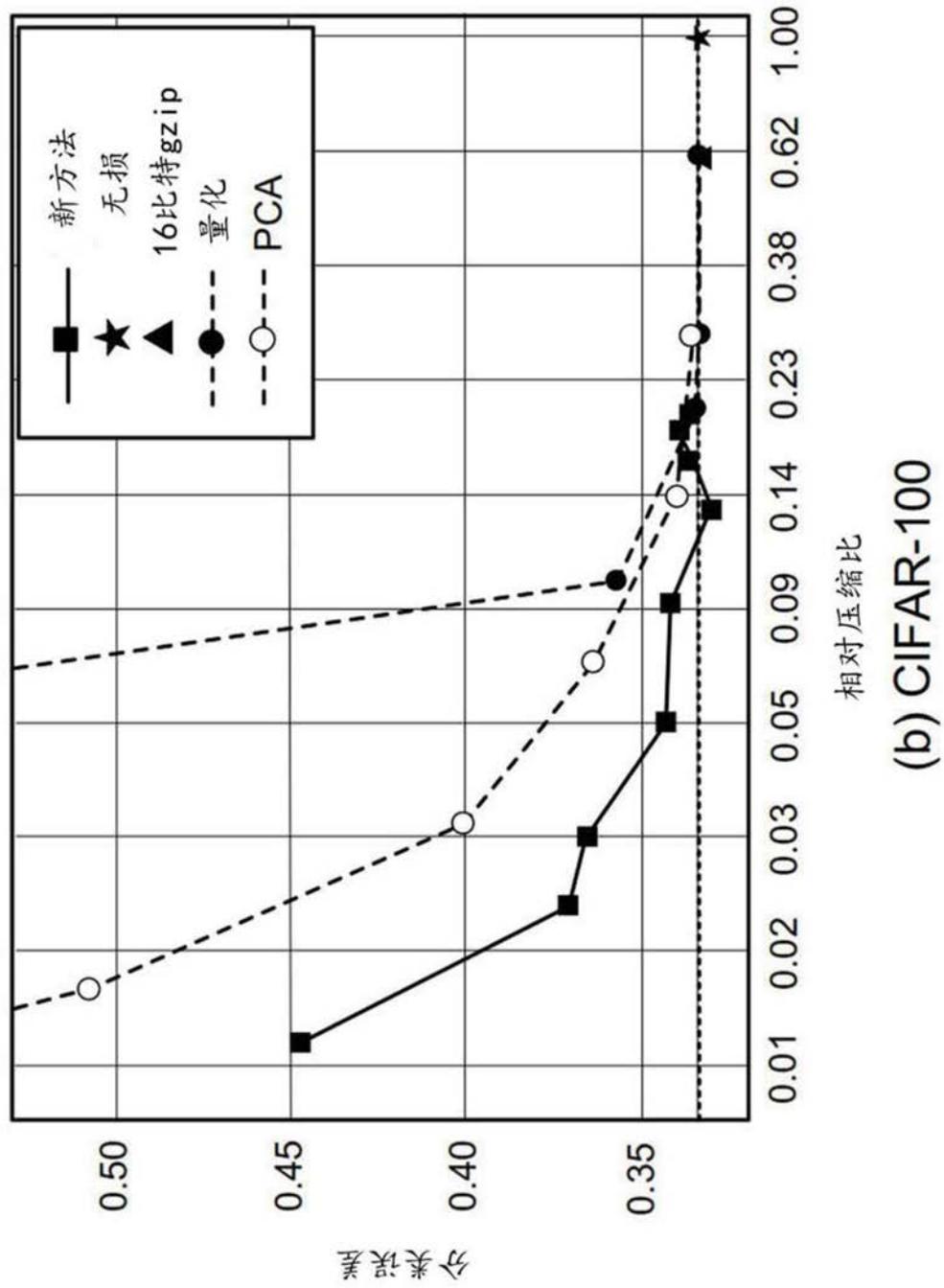
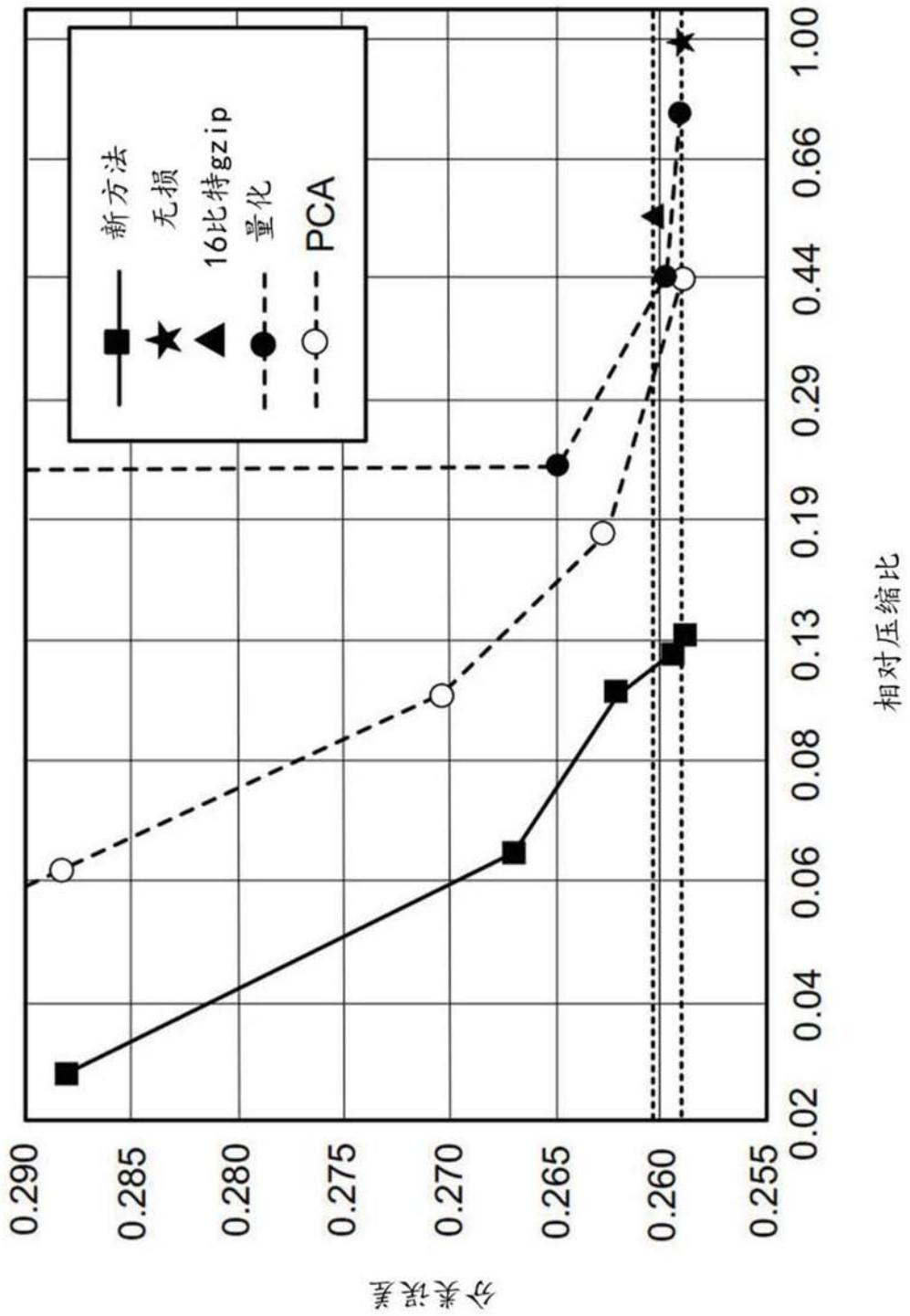


图3B

410 ↷



分类误差

相对压缩比

图4

510 ↗

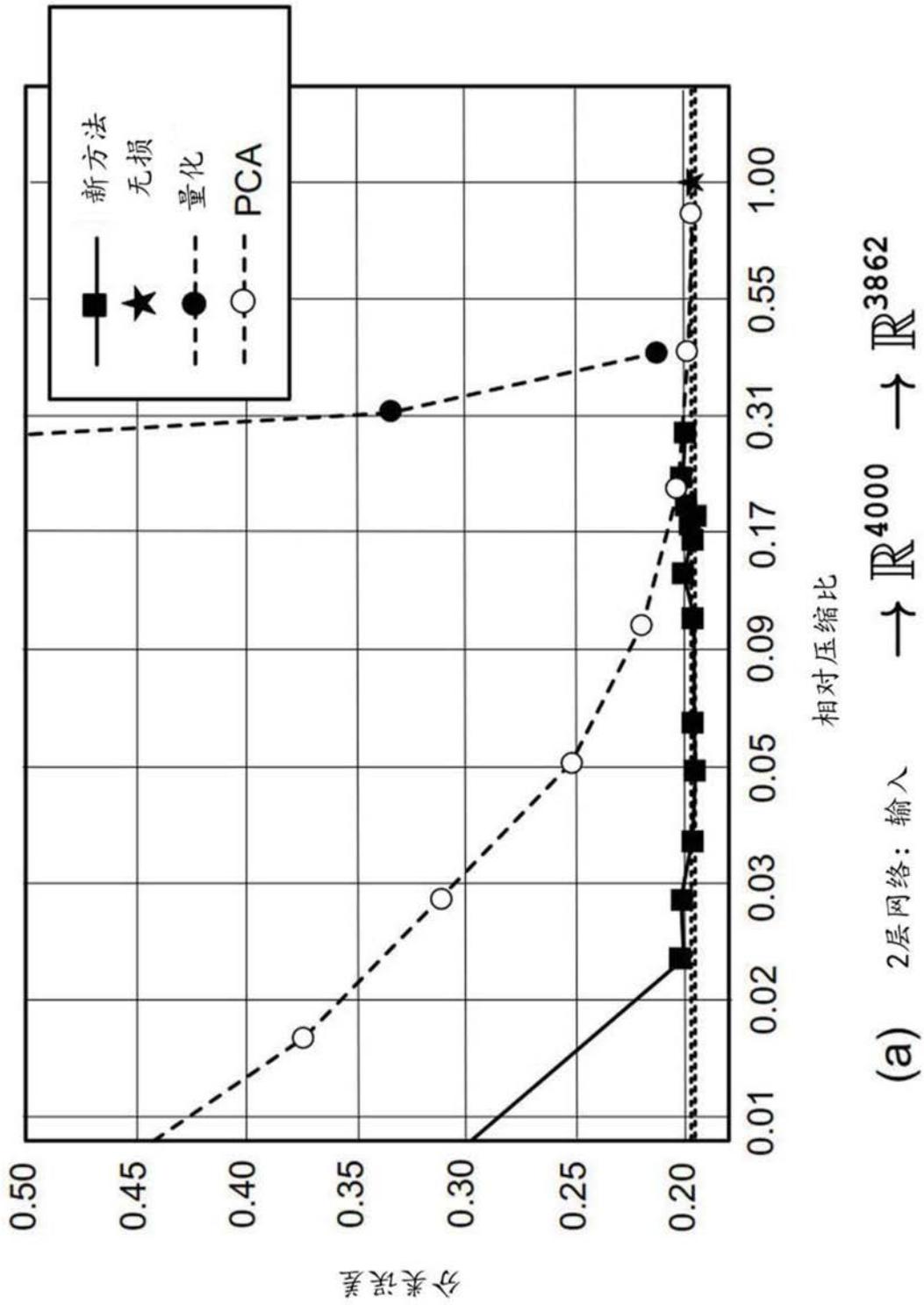


图5A

520 ↷

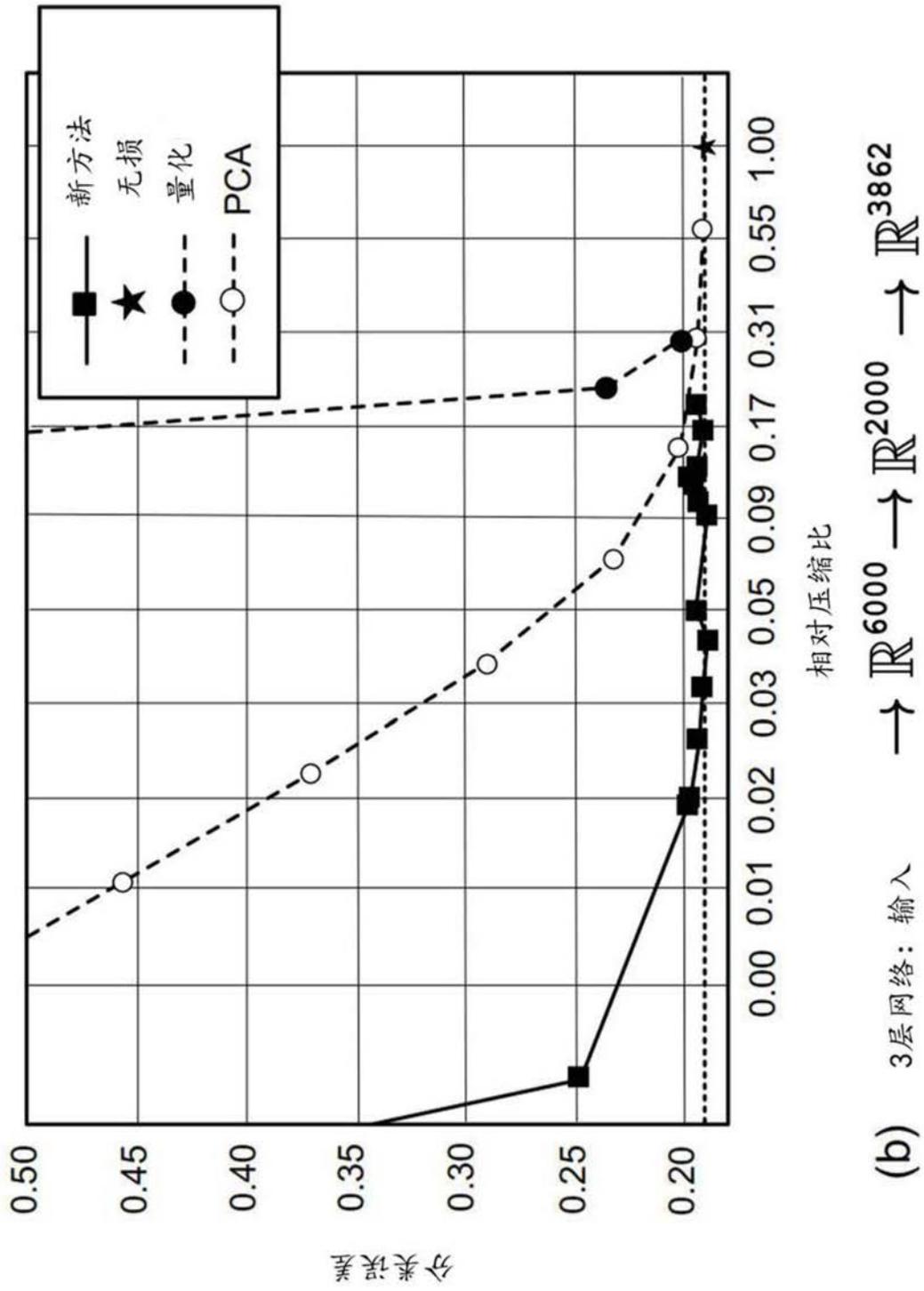


图5B