

基于点对相似度的深度非松弛哈希算法

汪海龙¹ 禹晶¹ 肖创柏¹

摘要 哈希学习能够在保持数据之间语义相似性的同时, 将高维数据投影到低维的二值空间中以降低数据维度实现快速检索. 传统的监督型哈希学习算法主要是将手工设计特征作为模型输入, 通过分类和量化生成哈希码. 手工设计特征缺乏自适应且独立于量化过程使得检索的准确率不高. 本文提出了一种基于点对相似度的深度非松弛哈希算法, 在卷积神经网络的输出端使用可导的软阈值函数代替常用的符号函数使准哈希码非线性接近 -1 或 1 , 将网络输出的结果直接用于计算训练误差, 在损失函数中使用 ℓ_1 范数约束准哈希码的各个哈希位接近二值编码. 模型训练完成之后, 在网络模型外部使用符号函数, 通过符号函数量化生成低维的二值哈希码, 在低维的二值空间中进行数据的存储与检索. 在公开数据集上的实验表明, 本文的算法能够有效地提取图像特征并准确地生成二值哈希码, 且在准确率上优于其他算法.

关键词 哈希学习, 非松弛, 卷积神经网络, 交叉熵

引用格式 汪海龙, 禹晶, 肖创柏. 基于点对相似度的深度非松弛哈希算法. 自动化学报, 2021, 47(5): 1077–1086

DOI 10.16383/j.aas.c180571

Deep Non-relaxing Hashing Based on Point Pair Similarity

WANG Hai-Long¹ YU Jing¹ XIAO Chuang-Bai¹

Abstract Hashing aims to project high-dimensional data into low-dimensional binary space and at the same time preserve the semantic similarity in the original space for fast retrieval. Traditional supervised hashing methods adopt hand-crafted features for hashing function learning and then generate hash codes by classification and quantization. The lack of adaptability and independence of the quantization procedure lead to low retrieval precision of supervised hashing methods with hand-crafted features in image retrieval. In this paper, we present a novel deep non-relaxation hashing method based on point pair similarity. In our method, a differentiable soft thresholding function is used to encourage hash-like codes to approach -1 or 1 nonlinearly at the output of the convolutional neural network instead of the widely used symbol function for quantization. The output of the soft thresholding function is directly used to compute the error of the network training, and a loss function is elaborately designed with ℓ_1 -norm to constrain each bit of hash-like codes to be as binary as possible. Finally, the symbol function is added outside the trained network model to generate binary hash codes for image storage and retrieval in the low-dimensional binary space. Extensive experiments on two large-scale public datasets show that our method can effectively learn image features and generate accurate binary hash codes, and outperform the state-of-the-art methods in terms of mean average precision.

Key words Hash learning, non-relaxation, convolutional neural network (CNN), cross entropy

Citation Wang Hai-Long, Yu Jing, Xiao Chuang-Bai. Deep non-relaxing hashing based on point pair similarity. *Acta Automatica Sinica*, 2021, 47(5): 1077–1086

近年来, 随着计算机软件和硬件技术的发展, 图像、视频等数据的维度和数量不断增加, 为了解决海量的高维数据的存储和检索问题, 研究学者提出了将高维数据投影到低维二值空间的哈希学习方法. 哈希学习方法是一种在保持图像或视频等高维数据

间相似性的条件下, 通过哈希函数或函数簇将高维空间的数据投影到低维汉明空间的二值编码的机器学习方法. 通过哈希学习方法对数据建立索引, 提高图像等高维数据的检索效率, 并节省存储空间. 现有的哈希方法大致可以分为两类: 数据独立的方法和数据依赖的方法^[1].

数据独立的方法使用随机投影来构造哈希函数. 局部敏感哈希 (Locality sensitive hashing, LSH) 算法^[2–4] 于 1998 年由 Indyk 等提出, 该算法在原始空间中使用随机线性投影将距离近的数据投影到相似的二值编码中. 该算法的哈希函数简单易实现, 计算速度快, 但准确率不高. 核化局部敏感哈希 (Kernelized locality-sensitive hashing, KLSH) 算法^[5] 对 LSH 算法提出了改进, KLSH 在核空间中随

收稿日期 2018-08-27 录用日期 2019-02-03
Manuscript received August 27, 2018; accepted February 3, 2019

北京市教育委员会科技计划 (KM201910005029), 北京市自然科学基金 (4212014) 资助

Supported by Scientific Research Common Program of Beijing Municipal Commission of Education (KM201910005029) and Beijing Municipal Natural Science Foundation (4212014)

本文责任编辑 张敏灵

Recommended by Associate Editor ZHANG Min-Ling

1. 北京工业大学信息学部 北京 100124

1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124

机构造哈希函数, 不需要考虑原始空间的数据分布, 可以选择任意核函数作为相似性度量函数. 但由于核函数的引入, 很大程度上增加了计算复杂度. 数据独立的方法虽然简单易实现, 但是由于这类方法不使用训练数据训练模型, 所以需要很长的哈希码才能达到较高的准确率. 数据依赖的方法使用训练数据来学习将高维数据投影为低维二值编码的哈希函数. 如同一般的机器学习方法, 数据依赖的方法可以分为非监督型方法和监督型方法.

非监督型哈希学习方法是数据没有类别信息或不利用类别信息情况下, 根据某种距离度量检索最近邻数据的方法. 谱哈希 (Spectral hashing, SH) 算法^[6] 是一种典型的非监督哈希算法, 其采用主成分分析 (Principal component analysis, PCA) 算法^[7] 获取图像数据的各个主成分方向, 并计算所有的解析特征函数 (Analytical eigenfunction), 然后使用符号函数将这些解析特征函数的值量化为二值编码. 由于在实际情况中很难同时满足谱哈希的两个约束条件, 即假设从多维均匀分布的空间中采样数据以及不同维度上的哈希编码之间相互独立, 因此谱哈希的使用受到一定限制. 迭代量化哈希 (Iterative quantization, ITQ) 算法^[8] 将数据投影到中心为零 (Zero-centered) 的二值超立方体 (Binary hypercube) 的顶点, 使得投影后的数据与原数据之间的量化误差能够最小化, 并使用一种简单有效的交替最小化 (Alternating minimization) 方法求解目标函数. 与 SH 算法相比, ITQ 算法不仅应用范围更广, 而且经实验验证 ITQ 算法的检索效果更好.

监督型哈希学习方法是指出将图像或其他高维数据的类别标签用于估计二值编码模型的输出, 在迭代过程中不断利用图像的标签信息计算实际输出与预期输出之间的误差反馈给模型, 不断调整模型参数的方法. 由于监督型哈希方法利用了数据的标签信息, 一般情况, 监督型哈希方法的效果比非监督性哈希方法的效果好. 2011 年, Norouzi 等提出了最小损失哈希 (Minimal loss hashing, MLH) 算法^[9], 这是一种典型的监督型哈希学习算法, 该算法通过结构化支撑向量机 (Structural support vector machine, Structural SVM) 建立损失函数, 由于损失函数不连续且非凸, 难以直接求解, 该算法通过构造并最小化损失函数的上界来求解损失函数. 基于核的监督哈希 (Kernel-based supervised hashing, KSH) 算法^[10] 采用核的形式来构造哈希函数, 使得其泛化可以处理非线性可分的问题, 并且, 不同于 MLH 算法的是, KSH 算法不直接通过最小化哈希码之间的汉明距离来学习哈希函数, 而是通过最小化哈希码的内积, 使问题变得更简单, 更容易求解. 2015 年, Shen 等^[11] 提出了监督型离散哈

希 (Supervised discrete hashing, SDH) 算法, 该算法使用 Lagrange 乘子法将离散约束问题松弛为凸优化问题, 在求解阶段使用循环坐标下降法 (Cyclic coordinate descent, DCC) 求解目标函数, 即每次迭代固定其他的哈希位, 只将一个哈希位作为变量求解, SDH 算法在保持准确率的同时, 提高了算法的效率.

上述传统的哈希学习方法在提取图像特征时采用手工设计特征, 例如梯度方向直方图 (Histogram of oriented gradient, HOG)、尺度不变特征变换 (Scale invariant feature transform, SIFT) 等, 不同场景的图像使用相同的手工设计特征在一定程度上影响了哈希学习算法的性能, 单一的特征导致检索准确率不高. 近年来, 卷积神经网络 (Convolutional neural network, CNN)^[12-13] 在计算机视觉领域成为主要的特征学习工具, 在图像分类^[14]、目标检测^[15-16] 等方向得到广泛应用. 目前许多学者也尝试将 CNN 用于哈希学习方法中, 陆续出现了一些深度哈希学习方法.

2014 年, Xia 等^[17] 提出了卷积神经网络哈希 (Convolutional neural network hashing, CNNH) 算法, 该算法利用深层框架来学习图像的特征, 然后以此特征学习哈希函数. 这是一种两阶段哈希学习算法, 第 1 阶段通过对相似度矩阵进行分解, 预生成样本的二值编码; 第 2 阶段将样本数据输入 CNN, 使用卷积层提取图像特征, 通过最小化网络输出值与第 1 阶段预编码之间的误差, 获得样本的哈希码. CNNH 相对于非深度哈希学习算法在准确率上有明显的提高, 然而也存在明显的问题, 第 1 阶段中相似度矩阵分解是一个非凸优化问题, 分解过程中需要大量的松弛方法, 导致准确率会受到很大的影响; 第 2 阶段仅利用了 CNN 自动提取图像特征, 没有参与第 1 阶段的计算. 基于监督型深度哈希的快速图像检索 (Deep supervised hashing for fast image retrieval, DSH) 算法^[18] 利用成对的图像和图像的标签作为网络的输入进行训练, 并联合使用对比损失函数 (Contrastive loss) 与哈希码的 l_1 范数正则项作为网络的损失函数, 以此约束哈希码, 使输出在汉明空间的哈希码更加多样化, 避免了两阶段哈希算法中第 1 阶段不能利用 CNN 网络, 以及使用 sigmoid 函数导致网络收敛速度过慢的问题. 2016 年, Li 等^[19] 提出了基于标签对的监督型深度哈希特征学习 (Feature learning based deep supervised hashing with pairwise labels, DPSH) 算法, 该算法通过图像的类别标签构造图像的标签对矩阵, 根据图像的标签对构建交叉熵损失函数, 以此衡量深度卷积神经网络训练的损失, 使用基于 Lagrange 乘子法的松弛优化方法, 对约束条件进行松弛, 去掉

符号函数的约束条件, 解决离散约束问题. 但是由于该算法使用 Lagrange 乘子, 某些哈希位会被过度松弛, 导致相似点对之间的语义信息保留不完整. 连续深度哈希学习 (Deep learning to hash by continuation, HashNet) 算法^[20] 也使用图像的标签对构建目标损失函数, 该算法在目标函数构造过程中引入加权似然函数, 有效地解决了相似点对数据不平衡问题, 同时在网络的输出端使用尺度双曲正切函数 (Scaled tanh function) 对网络输出进行阈值化, 并在迭代过程中, 不断增大尺度系数, 使其逼近符号函数的功能. 从实验结果来看, 与传统的哈希学习方法相比, 监督型深度哈希学习方法在准确率方面明显胜过传统的方法.

本文提出了一种基于点对相似度的深度非松弛哈希 (Deep non-relaxing hashing based on point pair similarity, DNRH) 算法, 该算法使用交叉熵保持相似点对之间的语义相似性, 在卷积神经网络的输出端使用一种软阈值函数阈值化网络输出得到准哈希码, 并使用 ℓ_1 范数约束输出端的准哈希码, 使得准哈希码的绝对值逼近 1, 避免了 Lagrange 乘子的松弛求解对模型准确率的影响. 本文利用卷积神经网络提取图像的特征表示并学习哈希函数生成哈希码, 这样就避免了相似度矩阵分解的不准确对后续量化过程的影响. 综上所述, 本文的主要贡献归纳如下:

1) 在损失函数中引入 ℓ_1 范数约束输出端的准哈希码, 通过正则项约束准哈希码, 使其各哈希位的绝对值逼近 1.

2) 为了减小量化误差, 在 CNN 的输出端使用软阈值函数, 使输出的准哈希码非线性地接近 -1 或 1 . 在模型外部使用符号函数 $\text{sgn}(\cdot)$, 将准哈希码量化为二值哈希码.

本文后续结构组织如下: 第 1 节介绍本文算法的背景, 第 2 节描述本文算法的模型、求解以及网络结构, 第 3 节通过实验验证本文算法的性能, 第 4 节为全文结论.

1 背景

1.1 问题描述

给定 n 幅图像的样本数据 $X = \{\mathbf{x}_i\}_{i=1}^n \in \mathbf{R}^{d \times n}$, 每幅输入图像为 $\mathbf{x}_i \in \mathbf{R}^d$. 其中, n 为图像的数量, d 为图像的尺寸. 输出数据为 $B \in \{-1, 1\}^{l \times n}$, B 的第 i 列 $\mathbf{b}_i \in \{-1, 1\}^l$ 表示第 i 个样本 \mathbf{x}_i 所对应的长度为 l 的二值哈希码, 简单来讲, 哈希学习旨在从训练数据中自动学习一个哈希函数 $H(\cdot)$, 一幅图像通过哈希函数可表示为 $\mathbf{b}_i = H(\mathbf{x}_i) = [h_1(\mathbf{x}_i), \dots, h_l(\mathbf{x}_i)]^T$. 对于线性哈希编码函数, 每一个哈希

函数 $h_i(\cdot)$ 将单幅图像投影为一个哈希位, l 个哈希函数组成一个函数簇将单幅图像投影成一个 l 位的二值哈希码.

在监督型哈希学习中, 每一个数据样本都有一个标签, 样本的标签矩阵为 $Y = \{\mathbf{y}_i\}_{i=1}^n \in \mathbf{R}^{C \times n}$, 其中, $\mathbf{y}_i \in \{0, 1\}^C$, C 为样本的总类别数. 每一个样本数据 \mathbf{x}_i 对应一个 \mathbf{y}_i , 当样本 \mathbf{x}_i 属于第 c 类时, \mathbf{y}_i 的第 c 位为 1, 其他位均为 0. 通过样本的标签矩阵 Y , 可以定义样本之间的相似度矩阵 $S \in \{0, 1\}^{n \times n}$, 对于任意两个样本 \mathbf{x}_i 与 \mathbf{x}_j , 若 \mathbf{x}_i 与 \mathbf{x}_j 属于同一类别, 则 $s_{ij} = 1$, 否则, $s_{ij} = 0$.

1.2 交叉熵损失函数

对于任意两个长度相等的哈希码 \mathbf{b}_i 和 \mathbf{b}_j , 将这两个哈希码的相似度 φ_{ij} 用它们的内积定义为

$$\varphi_{ij} = \varphi(\mathbf{b}_i, \mathbf{b}_j) = \frac{1}{2} \mathbf{b}_i^T \mathbf{b}_j \quad (1)$$

内积越大, 相似度越大, 使用 sigmoid 函数对相似度 φ_{ij} 进行非线性阈值化, 将其范围规范化到区间 $(0, 1)$, 可得

$$\sigma(\varphi_{ij}) = \frac{1}{1 + e^{-\varphi_{ij}}} \quad (2)$$

基于哈希码相似度的度量, Li 等^[19] 和 Zhang 等^[21] 利用交叉熵损失函数保持点对之间的相似度, 图像点对的哈希码与相似度之间的似然 $p(s_{ij}|B)$ 定义为

$$p(s_{ij}|B) = \begin{cases} \sigma(\varphi_{ij}), & s_{ij} = 1 \\ 1 - \sigma(\varphi_{ij}), & s_{ij} = 0 \end{cases} \quad (3)$$

式中, s_{ij} 表示样本对之间的相似度, B 表示样本数据对应的哈希码. 由该似然函数表明, 当哈希码 \mathbf{b}_i 与 \mathbf{b}_j 越相似, 即 $\sigma(\varphi_{ij})$ 越大, 对应的似然函数 $p(s_{ij}|B)$ 就越大; 当哈希码 \mathbf{b}_i 与 \mathbf{b}_j 越不相似, 即 $1 - \sigma(\varphi_{ij})$ 越大, 对应的似然函数 $p(s_{ij}|B)$ 仍越大. 极大似然估计可表示为

$$\max_B \prod_{s_{ij} \in S} p(s_{ij}|B) = \prod_{s_{ij} \in S} \sigma(\varphi_{ij})^{s_{ij}} (1 - \sigma(\varphi_{ij}))^{1-s_{ij}} \quad (4)$$

对式 (4) 中极大似然估计的目标函数取负对数即是交叉熵损失函数, 可表示为

$$J_1(B) = - \sum_{s_{ij} \in S} \log p(s_{ij}|B) = - \sum_{s_{ij} \in S} [s_{ij} \varphi_{ij} - \log(1 + e^{\varphi_{ij}})] \quad (5)$$

于是将极大似然估计转换为最小化交叉熵损失

函数, 建立如下的约束最优化问题:

$$\begin{aligned} \min_B & - \sum_{s_{ij} \in S} [s_{ij} \varphi_{ij} - \log(1 + e^{\varphi_{ij}})] \\ \text{s.t. } & \mathbf{b}_i = \text{sgn}[W^T \phi(\mathbf{x}_i; \boldsymbol{\theta}) + \mathbf{v}], i = 1, 2, \dots, n \end{aligned} \quad (6)$$

式中, W 表示全连接层的神经元参数, \mathbf{v} 表示偏移量, $\boldsymbol{\theta}$ 表示网络卷积层的参数集合, \mathbf{b}_i 表示二值哈希码, \mathbf{b}_i 中每一位量化到离散值 -1 或 1 , $\phi(\cdot)$ 表示网络提取的图像特征, $\text{sgn}(\cdot)$ 为符号函数, 若 $x > 0$, 则 $\text{sgn}(\cdot) = 1$; 否则 $\text{sgn}(\cdot) = -1$.

由于符号函数不连续, 上述问题是一个离散的最优化问题, 很难求解. 为了求解上述问题, Li 等^[19] 使用基于 Lagrange 乘子法的松弛优化方法, 对约束条件进行松弛, 简化约束条件中符号函数的离散约束, 使其成为凸优化问题, 其目标函数为

$$\begin{aligned} \min_B & - \sum_{s_{ij} \in S} [s_{ij} \varphi_{ij} - \log(1 + e^{\varphi_{ij}})] + \\ & \lambda \sum_{i=1}^n \|\mathbf{b}_i - [W^T \phi(\mathbf{x}_i; \boldsymbol{\theta}) + \mathbf{v}]\|_2^2 \\ \text{s.t. } & \mathbf{u}_i = W^T \phi(\mathbf{x}_i; \boldsymbol{\theta}) + \mathbf{v}, i = 1, 2, \dots, n \\ & \varphi_{ij} = \frac{1}{2} \mathbf{u}_i^T \mathbf{u}_j, i, j = 1, 2, \dots, n \end{aligned} \quad (7)$$

其中, \mathbf{u}_i 表示网络的直接输出结果. 该方法在每一次迭代过程中使用符号函数量化生成 \mathbf{b}_i , 这样会出现大量不可忽略的损失, 导致某些特征对应哈希位的约束变弱, 使得计算结果不准确, 这也是使用 Lagrange 乘子法对约束条件松弛导致某些哈希码过度松弛的问题.

2 深度非松弛哈希算法

2.1 模型的建立

本文利用深度卷积神经网络训练样本数据, 利用交叉熵保持样本对之间的语义相似性, 为了减少网络输出的准哈希码的量化误差, 本文使用 Liu 等^[18] 提出的 ℓ_1 范数对网络输出的准哈希码的分布进行约束

$$J_2(B) = \sum_{i=1}^n \|\mathbf{b}_i\|_1 \quad (8)$$

该正则项旨在使准哈希码 \mathbf{b}_i 的各个哈希位逼近两个离散值 -1 或 1 , 即 \mathbf{b}_i 中每一位的绝对值越接近 1 时, 损失越小.

在迭代过程中直接使用网络的输出作为准哈希码有很大概率过度偏离 -1 或 1 , 这样将增大式 (8) 中 $J_2(B)$ 的损失值, 尽管符号函数能够很好地对其进行量化, 然而符号函数不可导. 相对于符号函数的

离散编码, 双曲正切函数能够使准哈希码的各哈希位非线性地接近 -1 或 1 , 且具有连续无限可导的良好性质, 其形式为

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (9)$$

Cao 等^[20] 使用尺度双曲正切函数 $\tanh(\beta x)$ 阈值化网络输出值为准哈希码, 其中 β 为尺度系数, 控制函数值逼近 1 或 -1 的速度. 类似地, 在双曲正切函数的基础上, 本文使用如下形式的函数:

$$\text{soft}(x) = \frac{1 - e^{-\eta x}}{1 + e^{-\eta x}} \quad (10)$$

式中, η 控制阈值函数的斜率, 本文中称其为软阈值函数.

与 HashNet 算法^[20] 不同的是, HashNet 算法在训练的过程中, 逐渐增大阈值函数的尺度系数, 随着迭代次数的增加, 阈值函数不断逼近并最终收敛至符号函数, 而本文的算法中软阈值系数 η 为模型参数, 通过实验获取最优参数后, 在每次迭代中, η 保持不变, 这样将简化模型并使模型快速收敛. 本文在网络输出端使用式 (10) 的软阈值函数, 离散约束问题就转化为可导损失函数对模型参数求导的问题. 在迭代过程中, 软阈值函数将网络输出的结果平滑地非线性映射到 $(-1, 1)$ 之间, 并且将大部分哈希位阈值化到 -1 和 1 两个值附近. 图 1 给出一维空间 $\text{soft}(x)$ 函数的曲线, 随着 η 的增大, 软阈值函数 $\text{soft}(x)$ 的函数值更加迅速地集中靠近离散值 -1 或 1 .

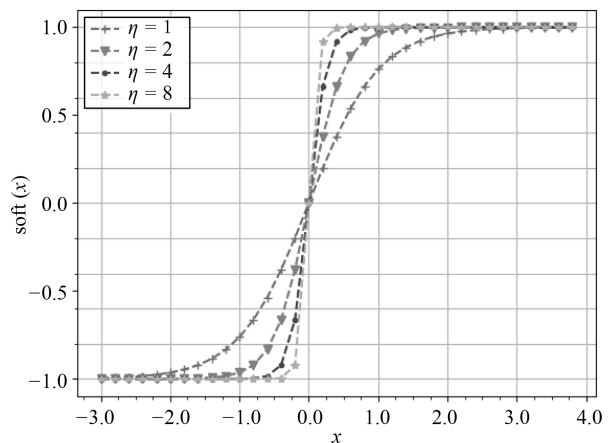


图 1 $\text{soft}(x)$ 的函数曲线

Fig. 1 The function curve of the $\text{soft}(x)$

结合式 (5) 的交叉熵损失函数和式 (8) 的正则项, 并在卷积网络输出端使用式 (10) 的软阈值函数 $\text{soft}(x)$, 本文建立如下目标函数:

$$\begin{aligned} \min_B J(B) = & - \sum_{s_{ij} \in S} [s_{ij} \varphi_{ij} - \log(1 + e^{\varphi_{ij}})] + \\ & \lambda \sum_{i=1}^n \|\mathbf{b}_i\| - 1 \\ \text{s.t. } \mathbf{b}_i = & \text{soft}[W^T \phi(\mathbf{x}_i; \boldsymbol{\theta}) + \mathbf{v}], \quad i = 1, 2, \dots, n \\ \varphi_{ij} = & \frac{1}{2} \mathbf{b}_i^T \mathbf{b}_j, \quad i, j = 1, 2, \dots, n \end{aligned} \quad (11)$$

式中, n 表示样本数, $s_{ij} \in \{0, 1\}$ 表示样本 i 和样本 j 是否相似, λ 表示正则项系数, $\text{soft}(\cdot)$ 表示软阈值函数, 对向量逐元素运算, η 表示软阈值函数的控制参数, \mathbf{b}_i 表示前向网络输出的准哈希码, φ_{ij} 表示两个哈希码之间的相似度. 目标函数中, 第 1 项为语义保真项, 第 2 项为准哈希码的正则项. 在本文的网络模型输出端使用 $\text{soft}(\cdot)$, 输出结果 \mathbf{b}_i 将迅速逼近 -1 和 1 这两个值, 使得 ℓ_1 范数正则项损失减小, 同时加快网络收敛的速度. 网络经过训练后, 在网络模型外部使用符号函数将准哈希码量化为二值哈希码.

2.2 参数学习

参数变量 W 和 \mathbf{v} 通过神经网络的反向传播 (Back propagation, BP) 求解, 在每一次迭代中, 通过随机梯度下降算法 (Stochastic gradient descent, SGD) 更新 W 和 \mathbf{v} . 目标函数用两项之和表示为 $J = J_1 + \lambda J_2$, 其中, $J_1 = - \sum_{s_{ij} \in S} [s_{ij} \varphi_{ij} - \log(1 + e^{\varphi_{ij}})]$, $J_2 = \sum_{i=1}^n \|\mathbf{b}_i\| - 1$.

1) 对准哈希码 \mathbf{b}_i 求偏导

第 1 项 J_1 对 \mathbf{b}_i 求偏导, 可得:

$$\begin{aligned} \frac{\partial J_1}{\partial \mathbf{b}_i} = & \frac{1}{2} \sum_{j: s_{ij} \in S} (a_{ij} - s_{ij}) \mathbf{b}_j + \\ & \frac{1}{2} \sum_{j: s_{ji} \in S} (a_{ji} - s_{ji}) \mathbf{b}_j \end{aligned} \quad (12)$$

其中, $a_{ij} = \sigma(\frac{1}{2} \mathbf{b}_i^T \mathbf{b}_j)$.

第 2 项 J_2 对 \mathbf{b}_i 求偏导, 可得:

$$\frac{\partial J_2}{\partial \mathbf{b}_i} = \delta(\mathbf{b}_i) \quad (13)$$

其中, $\delta(x) = \begin{cases} 1, & -1 \leq x \leq 0 \text{ 或 } x \geq 1 \\ -1, & \text{否则} \end{cases}$.

结合式 (12) 和式 (13) 可得损失函数 J 对 \mathbf{b}_i 的偏导数为

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{b}_i} = & \frac{1}{2} \sum_{j: s_{ij} \in S} (a_{ij} - s_{ij}) \mathbf{b}_j + \\ & \frac{1}{2} \sum_{j: s_{ji} \in S} (a_{ji} - s_{ji}) \mathbf{b}_j + \lambda \delta(\mathbf{b}_i) \end{aligned} \quad (14)$$

2) 对参数 W 求偏导

设 $\mathbf{z}_i = W^T \phi(\mathbf{x}_i; \boldsymbol{\theta}) + \mathbf{v}$, 损失函数 J 对 W 求偏导, 根据链式求导法则可推导出:

$$\frac{\partial J}{\partial W} = \left(\frac{\partial \mathbf{b}_i}{\partial \mathbf{z}_i} \right)^T \phi(\mathbf{x}_i; \boldsymbol{\theta}) \left(\frac{\partial J}{\partial \mathbf{b}_i} \right)^T \quad (15)$$

其中, $\frac{\partial J}{\partial \mathbf{b}_i}$ 在式 (14) 中求解, 根据 $\mathbf{b}_i = \text{soft}(\mathbf{z}_i)$, $\frac{\partial \mathbf{b}_i}{\partial \mathbf{z}_i}$ 的计算式为

$$\frac{\partial \mathbf{b}_i}{\partial \mathbf{z}_i} = \text{diag} \{ \text{soft}'(\mathbf{z}_i) \} \quad (16)$$

其中, $\text{diag}\{\cdot\}$ 表示主对角线上为向量元素的对角矩阵, $\text{soft}'(x) = (2\eta e^{-\eta x}) / (1 + e^{-\eta x})^2$, 对向量逐元素运算.

因此, 可得损失函数 J 对 W 的偏导数为

$$\begin{aligned} \frac{\partial J}{\partial W} = & (\text{diag} \{ \text{soft}'(\mathbf{z}_i) \})^T \phi(\mathbf{x}_i; \boldsymbol{\theta}) \left(\frac{\partial J}{\partial \mathbf{b}_i} \right)^T = \\ & [\text{soft}'(\mathbf{z}_i) \odot \phi(\mathbf{x}_i; \boldsymbol{\theta})] \left(\frac{\partial J}{\partial \mathbf{b}_i} \right)^T \end{aligned} \quad (17)$$

其中, \odot 为哈达玛积 (Hadamard product).

3) 对偏移量 \mathbf{v} 求偏导

损失函数 J 对 \mathbf{v} 求偏导, 根据链式求导法则可得:

$$\frac{\partial J}{\partial \mathbf{v}} = \left(\frac{\partial \mathbf{b}_i}{\partial \mathbf{z}_i} \frac{\partial \mathbf{z}_i}{\partial \mathbf{v}} \right)^T \frac{\partial J}{\partial \mathbf{b}_i} \quad (18)$$

其中, $\frac{\partial J}{\partial \mathbf{b}_i}$ 和 $\frac{\partial \mathbf{b}_i}{\partial \mathbf{z}_i}$ 分别在式 (14) 和式 (16) 中求解, $\frac{\partial \mathbf{z}_i}{\partial \mathbf{v}}$ 的计算式为

$$\frac{\partial \mathbf{z}_i}{\partial \mathbf{v}} = I_l \quad (19)$$

其中, $I_l \in \mathbf{R}^{l \times l}$ 为单位矩阵.

因此, 可得损失函数 J 对 \mathbf{v} 的偏导数:

$$\frac{\partial J}{\partial \mathbf{v}} = (\text{diag} \{ \text{soft}'(\mathbf{z}_i) \})^T \frac{\partial J}{\partial \mathbf{b}_i} = \text{soft}'(\mathbf{z}_i) \odot \frac{\partial J}{\partial \mathbf{b}_i} \quad (20)$$

由以上求解过程可知, 损失函数可导, 即反向传播的过程偏导数存在, 网络模型经过一定迭代能够收敛.

2.3 网络结构与参数

本文采用 AlexNet 网络模型作为深度哈希学习的基础模型, AlexNet 模型一共分为 8 层, 5 个卷积层和 3 个全连接层^[12]. 在每一个卷积层中使用线性整流函数 (Rectified linear unit, ReLU) 作为激励函数以及局部响应归一化 (Local response normalization, LRN) 处理数据, 然后经过下采样 (Pooling), 自动提取图像的特征表示, 再经过全连

接层输出结果. 本文使用的 AlexNet 网络模型及参数如图 2 所示, 第 1, 3, 5, 6, 7 层为卷积层. 卷积层均使用保留边界的滑动方式, 第 1 和第 3 层使用的卷积核的尺寸分别为 11×11 和 5×5 , 其他 3 个卷积核的尺寸为 3×3 , 第 1 层的滑动步长水平垂直方向均为 2, 其他 4 个卷积层的滑动步长水平垂直方向均为 1. 第 2, 4, 8 层为下采样层, 使用非保留边界的滑动方式, 窗口尺寸为 3×3 , 滑动步长水平和垂直方向均为 2, 后三层为全连接层, 输出端使用软阈值函数阈值化输出为准哈希码.

2.4 算法流程

算法 1 给出了本文算法的伪代码, 输入数据为一定格式的图像, 通过 CNN 的卷积层和池化层提取图像的特征表示, 将图像的特征表示输入 CNN 全连接层后使用软阈值函数得到图像的准哈希码. 根据目标函数, 使用反向传播算法更新参数 W 和 v , 通过指定次数的迭代训练完成模型, 最后, 在训练完成的模型外部使用符号函数, 对图像的准哈希码量化输出二值哈希码.

算法 1. 深度非松弛哈希学习算法 (DNRH)

输入. n 幅图像数据集: $X = \{\mathbf{x}_i\}_{i=1}^n \in \mathbf{R}^{d \times n}$

输出. n 幅图像对应的哈希码: $B \in \{-1, 1\}^{l \times n}$ 及网络参数

初始化. 高斯分布初始化权重 W 和偏移量 v

迭代过程.

1. 前向网络计算对应的 B
2. 利用式 (11) 损失函数计算训练误差
3. 使用随机梯度下降算法, 通过式 (17) 和式 (18) 更新

参数 W 和 v

停止. 达到固定的最大迭代次数停止

返回. 网络参数 W 和 v , 网络模型外部使用 $\text{sgn}(\cdot)$, 前向传播计算哈希码 B .

3 实验结果与分析

3.1 在 CIFAR-10 和 NUS-WIDE 上的实验

将本文提出的算法与各种流行的哈希学习算法

进行比较, 其中包括 2 个非监督哈希学习算法 SH^[6] 和 ITQ^[8], 4 个监督型非深度哈希学习算法 MLH^[9]、KSH^[10]、SDH^[11] 和 BRE^[22], 以及 6 个监督型深度哈希学习算法 CNNH^[17]、FP-CNNH^[23]、NINH^[24]、DHN^[25]、DSH^[18] 和 DPSH^[19]. 在 6 个非深度哈希学习算法中从图像中提取 512 维的 GIST 特征向量作为哈希函数的输入. 本文的 DNRH 算法和其他 6 个深度哈希学习算法 CNNH、DHN、FP-CNNH、NINH、DSH、DPSH 根据图像的尺寸使用 AlexNet 网络卷积层自动提取一定维度的图像特征作为哈希函数的输入, 生成哈希码.

为了将本文提出的 DNRH 算法与上述哈希学习算法的性能比较, 实验中, 本文选取 CIFAR-10^[26] 和 NUS-WIDE^[27] 两个公开的数据集. 在这两个数据集上训练模型并进行测试. CIFAR-10 由 60 000 幅 32×32 像素的 RGB 彩色图像构成, 共包含飞机、汽车、鸟、猫、鹿、狗、青蛙、马、船和卡车 10 个类别^[26]. 其中 50 000 幅作为训练数据, 10 000 幅作为测试数据. NUS-WIDE 数据集是由新加坡国立大学媒体搜索实验室创建的网络图像数据集, 共有 5 018 个标签, 81 个类别, 将近 270 000 幅来源于网络的图像. 与 CIFAR-10 不同的是, NUS-WIDE 数据集中某一幅图像可能赋予一个或者多个标签, 在本文实验中, 主要抽取动物、建筑、植物、夕阳、人、天空、云、水、草、窗户、湖、树、汽车、沙滩、鸟、狗、马、雪、花、海洋、马路 21 个常用的类别^[27] 用于实验, 每个类别中至少包括 5 000 幅图像.

在实验阶段, 在 CIFAR-10 数据集中, 本文随机从每一个类别中抽取 500 幅图像作为训练数据, 100 幅图像作为测试数据, 一共 5 000 幅图像组成训练集, 1 000 幅图像组成测试集. 在 NUS-WIDE 数据集中, 随机从每一个类别选出 500 幅图像作为训练数据, 100 幅图像作为测试数据. 一共 10 500 幅图像组成训练集, 2 100 幅图像组成测试集. 在 NUS-WIDE 数据集上直接使用 AlexNet 网络模型, 将每幅图像调整为 224×224 像素的尺寸以适应 AlexNet 网络模型的输入. 为了验证算法的鲁棒性,

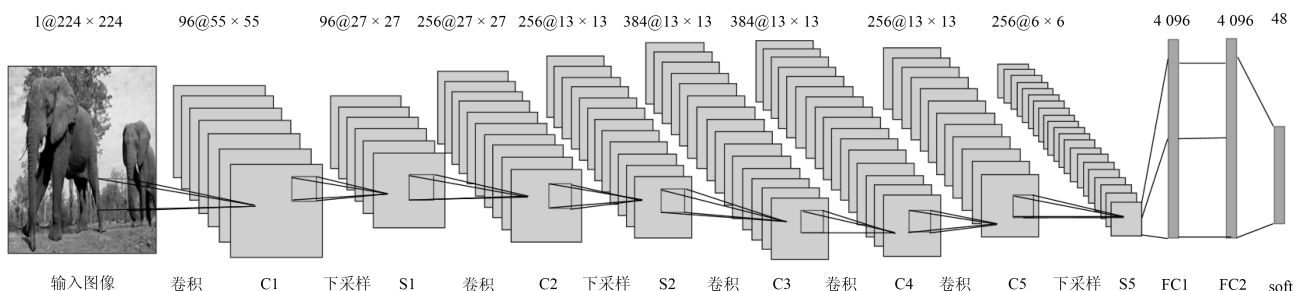


图 2 本文算法使用的网络模型

Fig. 2 The network model of our algorithm

本文在 CIFAR-10 和 NUS-WIDE 两个数据集上 ℓ_1 范数正则项系数 λ 均设置为 0.05, 软阈值函数控制参数 η 均设置为 12.

为了避免实验误差以及随机性, 在评价哈希学习算法的有效性上, 本文采用平均准确率 (Mean average precision, MAP)^[28] 作为检索图像的评价指标. 选取一定长度的哈希码后, 在测试集中, 选取一部分图像作为待检索的样本, 依次计算待检索的图像与数据集中其他图像的汉明距离, 根据汉明距离将图像排序并计算排序列表中待检索图像类别相同的图像数量与检索图像总数之间的比值, 作为准确率.

表 1 比较了 CIFAR-10 数据集上本文的 DNRH 算法与现有的哈希学习算法的 MAP. 从表 1 可以观察到, 在 4 种长度的哈希码上, 本文的 DNRH 算法的平均准确率均明显高于其他所有算法. 通过比较 CNNH、DHN、FP-CNNH、NINH、DPSH、DSH 和 DNRH 等 7 个深度哈希算法以及其他 6 个非深度哈希算法的 MAP 可知, 深度哈希算法高于非深度哈希算法的平均准确率, 这表明使用 CNN 模型的深度哈希学习算法自动提取图像特征表示比传统手工提取图像特征表示具有更好的性能. 对于 12 bit 的哈希码, 所有算法的检索准确率相对偏低, 随着哈希码长度的增加, 所有算法检索准确率都逐渐提高, 哈希码长度为 48 bit 时, 所有算法的检索准确率达到最高. 相比于 48 bit 以下的哈希码, 使用 48 bit 的哈希码可以存储更多的图像特征, 在检索时, 可以利用更多的图像特征, 取得更高的准确率.

表 1 各种算法在 CIFAR-10 上的 MAP

Table 1 The MAP of different algorithms on CIFAR-10

算法	12 bit	24 bit	32 bit	48 bit
DNRH	0.726	0.749	0.753	0.768
DPSH	0.713	0.727	0.744	0.757
DSH	0.616	0.651	0.661	0.676
DHN	0.555	0.594	0.603	0.621
FP-CNNH	0.612	0.639	0.625	0.616
NINH	0.552	0.566	0.558	0.581
CNNH	0.439	0.511	0.509	0.532
SDH	0.285	0.329	0.341	0.356
KSH	0.316	0.390	0.412	0.458
MLH	0.182	0.195	0.207	0.211
BRE	0.159	0.181	0.193	0.196
ITQ	0.162	0.169	0.172	0.175
SH	0.127	0.128	0.126	0.129

NUS-WIDE 数据集的图像相对于 CIFAR-10 数据集的图像具有更高的像素, 更完整的图像细节, 更贴近实际应用中的图像. 在 NUS-WIDE 数据集中, 一幅图像可能包含多个标签, 在检索过程中, 只要检索到的图像与待检索图像包含有相同的标签, 就判定为正确检索. 表 2 比较了在 NUS-WIDE 数据集上本文的 DNRH 算法与现有的哈希学习算法在不同长度哈希码上的平均准确率. 由于 NUS-WIDE 数据集的图像数量很大, 在该数据集上, 本文用每个测试样本检索返回的前 5 000 个样本计算 MAP. 在相同长度的哈希码上, 本文 DNRH 算法在 12 bit、24 bit、32 bit、48 bit 上的平均准确率分别为 0.769、0.792、0.804、0.814, 均高于其他的哈希学习算法, 证明了本文的算法的普适性. 其中, 对于 DPSH 算法, 本文仍然使用在 CIFAR-10 数据集上实验的 Lagrange 乘子 $\eta = 10$, 并在各种算法使用相同的训练集和测试集的设置下, 在 NUS-WIDE 数据集上重新运行作者提供的代码, 计算其平均准确率. 随着哈希码长度的增加, 几乎所有算法的平均检索准确率都有一定程度的提高, 尤其 SDH 算法, 48 bit 的哈希码对应的平均准确率相对于 12 bit 的哈希码的平均准确率提高了近 7%, 表明更多的哈希位能够表示更多的图像特征, 提高检索准确率.

表 2 各种算法在 NUS-WIDE 上的 MAP

Table 2 The MAP of different algorithms on NUS-WIDE

算法	12 bit	24 bit	32 bit	48 bit
DNRH	0.769	0.792	0.804	0.814
DPSH	0.747	0.788	0.792	0.806
DSH	0.548	0.551	0.558	0.562
DHN	0.708	0.735	0.748	0.758
FP-CNNH	0.622	0.628	0.631	0.625
NINH	0.674	0.697	0.713	0.715
CNNH	0.618	0.621	0.619	0.620
SDH	0.568	0.600	0.608	0.637
KSH	0.556	0.572	0.581	0.588
MLH	0.500	0.514	0.520	0.522
BRE	0.485	0.525	0.530	0.544
ITQ	0.452	0.468	0.472	0.477
SH	0.454	0.406	0.405	0.400

3.2 ℓ_1 范数和软阈值函数的约束效果

在本文提出的 DNRH 算法中, 软阈值函数的作用是在模型的前向计算中直接阈值化网络输出端的

结果, 而 l_1 范数作为目标函数的正则项在模型的反向传播中约束准哈希码, 使准哈希码各个位的绝对值逼近 1, 这两个模块的作用均为约束准哈希码. 为了验证联合使用 l_1 范数和软阈值函数的约束性能, 本文在 CIFAR-10 数据集上分别对 l_1 范数正则项独立约束、软阈值函数独立约束以及 l_1 范数和软阈值函数联合约束进行了实验.

表 3 列出了在 4 种长度的哈希码上, 不同模型对应的平均准确率, 其中, “交叉熵 + 软阈值” 表示使用式 (5) 的损失函数, 并在网络的输出端使用软阈值函数的模型, “交叉熵 + l_1 范数” 表示使用式 (11) 的损失函数, 并略去约束条件, 即在网络的输出端不使用软阈值函数的模型, “交叉熵 + l_1 范数 + 软阈值” 表示本文的 DNRH 算法模型, 即联合使用 l_1 范数和软阈值函数. 观察表 3 可知, “交叉熵 + l_1 范数” 和 “交叉熵 + 软阈值” 这两个模型的平均准确率明显低于 DPSH 算法, 表明单独使用 l_1 范数和软阈值函数的效果并不如以 Lagrange 乘子松弛求解的 DPSH 算法. 而联合使用 l_1 范数和软阈值函数 (交叉熵 + l_1 范数 + 软阈值) 在 4 种长度哈希码长度上, 其 MAP 相比于单独使用其中一个模块均提高了近 10%, 并且高于 DPSH 算法. 因此可得出结论, 联合使用 l_1 范数和软阈值函数能够更好地约束哈希码, 提升算法性能.

表 3 l_1 范数和软阈值函数约束在 CIFAR-10 上的 MAP
Table 3 The MAP of l_1 -norm and soft threshold function constraint on CIFAR-10

算法	12 bit	24 bit	36 bit	48 bit
交叉熵 + l_1 范数 + 软阈值	0.726	0.749	0.753	0.768
DPSH	0.713	0.727	0.744	0.757
交叉熵 + 软阈值	0.613	0.636	0.662	0.688
交叉熵 + l_1 范数	0.606	0.621	0.660	0.671

3.3 参数影响的分析

3.3.1 正则项系数 λ 对准哈希码分布的影响

为了检验 l_1 范数正则项对全连接层输出的准哈希码的约束能力, 本文在 CIFAR-10 数据集上对输出的准哈希码的分布情况进行了统计. 统计准哈希码中每一位的绝对值相对于 1 的距离分别在区间 $[0, 0.1)$, $[0.1, 0.2)$, $[0.2, 0.3)$, $[0.3, 0.4)$ 的概率, 如图 3 所示, 不同颜色表示不同的分布区间, 横轴表示正则项系数 λ , 纵轴表示哈希位落在不同区间的概率.

从图 3 中准哈希码各哈希位的分布情况可以看出, 随着 λ 的增大, 准哈希码各哈希位的绝对值更集中靠近 1, 尤其在不使用 l_1 范数 ($\lambda = 0$) 约束的情

况下, 准哈希码的哈希位在 $0 \sim 0.4$ 之间分布相对均匀, 这样在最后的量化过程中损失会增加, 导致结果不准确. 在目标函数中, 语义保真项用于保持点对之间的相似性, l_1 范数正则项用于约束准哈希码的分布, 正则项系数 λ 过大将过分增大 l_1 范数正则项的比重, 从而减小语义保真项的作用, 影响分类效果. 由此可以看出, 适当的 l_1 范数正则项对准哈希码的分布有很好的约束作用.

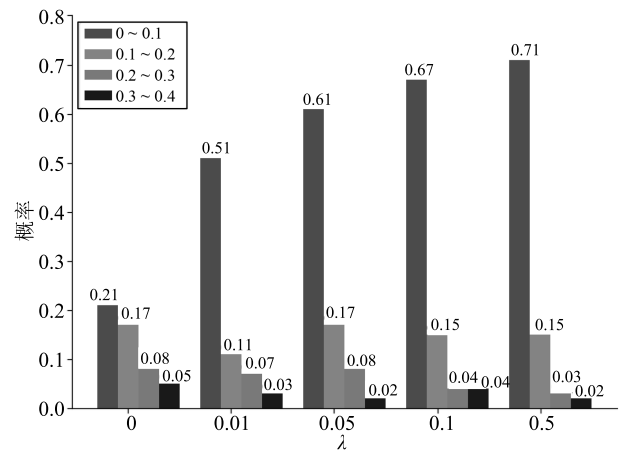


图 3 不同正则项系数 λ 下准哈希码的分布
Fig. 3 The distribution of hash code with different regularization coefficient λ

3.3.2 正则项系数 λ 对准确率的影响

正则项系数 λ 的取值, 不仅影响模型输出的准哈希码各哈希位的分布, 而且影响本文的 DNRH 算法训练模型的准确率. 表 4 给出了哈希码的长度取 48 时, 在 CIFAR-10 和 NUS-WIDE 数据集上 λ 取不同值时的平均准确率.

表 4 λ 的不同取值对应的 MAP
Table 4 The MAP on different λ

λ	CIFAR-10 数据集	NUS-WIDE 数据集
0	0.635	0.661
0.01	0.720	0.773
0.05	0.768	0.814
0.1	0.742	0.756
0.5	0.695	0.680

从表 4 可以看出, 在两个数据集上 λ 的取值对 MAP 的影响分布相同. 当 $\lambda = 0.05$ 时, 测试集上的检索效果最好, λ 取值过小或过大都会对检索的 MAP 造成影响. 分析其原因可知, λ 值过小, 目标函数对准哈希码分布的约束变弱, 模型输出的准哈希码的部分哈希位将过度偏离 -1 或 1 , 造成最终量

化为哈希码时损失增大; λ 的取值过大将造成目标函数中语义保真项的比重减小, 由于该约束项的作用是保持图像之间的相似语义, 减小该约束项的比重会造成相同类别之间的距离增大或者不同类别之间的距离减小, 即图像之间的相似性约束变弱, 使得检索效果变差。

3.3.3 软阈值函数控制参数 η 对准哈希码分布的影响

为了验证软阈值函数对准哈希码的阈值化效果, 本文在 NUS-WIDE 数据集上进行实验, 统计软阈值函数控制参数 η 取不同值时, 模型输出的准哈希码的分布情况。同样统计准哈希码中每一位的绝对值相对于 1 的距离分别在区间 $[0, 0.1)$, $[0.1, 0.2)$, $[0.2, 0.3)$, $[0.3, 0.4)$ 的概率, 如图 4 所示, 横轴表示软阈值函数控制参数 η 的取值, 纵轴表示哈希位落在不同区间的概率。

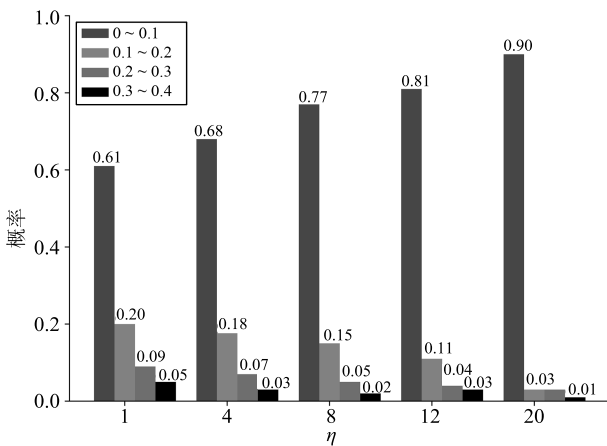


图 4 参数 η 取不同值时准哈希码的分布

Fig. 4 The distribution of hash code with different η

由图 4 可以看出, η 的取值越大, 准哈希码的各哈希位的越逼近 1 或 -1, 尤其在 $\eta = 20$ 时, 准哈希码误差在 0.1 以内的比例达到了 90%, 但是 η 的取值过大也会带来严重的问题, 当 $\eta = 20$ 时, 在模型的训练过程中, 损失函数始终振荡难以收敛, 这是因为当 η 取值过大时, 软阈值函数趋于不可导。为了在模型训练中使损失平稳收敛, 并且使准哈希码绝对值逼近 1, 经过多次实验, 本文中 η 的取值为 12。

4 结论

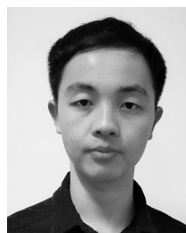
本文提出了一种基于点对相似度的深度非松弛哈希学习算法, 该算法通过 CNN 自动提取图像特征, 在目标函数中使用交叉熵保持相似点对之间的语义相似性, 在 CNN 输出端使用可导的软阈值函数代替传统方法中的符号函数, 使准哈希码非线性地接近 -1 或 1。并在损失函数中引入 ℓ_1 范数约束

输出的准哈希码, 使得准哈希码逼近二值编码。最终在训练好的模型外部使用符号函数, 将准哈希码量化为二值哈希码。在公开数据集 (CIFAR-10、NUS-WIDE) 上的实验验证了本文提出的 DNRH 算法优于其他哈希学习算法, 取得了令人满意的效果。

References

- Li Wu-Jun, Zhou Zhi-Hua. Learning to hash for big data: current status and future trends. *Chinese Science Bulletin*, 2015, **60**(5-6): 485-490 (李武军, 周志华. 大数据哈希学习: 现状与趋势. 科学通报, 2015, **60**(5-6): 485-490)
- Raginsky M, Lazebnik S. Locality-sensitive binary codes from shift-invariant kernels. In: *Proceedings of the 2009 Neural Information Processing Systems*. Vancouver, BC, Canada: Curran Associates, Inc, 2009. 1509-1517
- Indyk P, Motwani R. Approximate nearest neighbor: towards removing the curse of dimensionality. In: *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*. New York, NY, USA: ACM, 1998. 604-613
- Datar M, Immorlica N, Indyk P, Mirrokni V. Locality-sensitive hashing scheme based on p-stable distributions. In: *Proceedings of the 20th Annual Symposium on Computational Geometry*. New York, NY, USA: ACM, 2004. 253-262
- Kulis B, Grauman K. Kernelized locality-sensitive hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, **34**(6): 1092-1104
- Weiss Y, Torralba A, Fergus R. Spectral hashing. In: *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems*. British Columbia, Canada: MIT, 2008. 1753-1760
- Wold S, Esbensen K, Geladi P. Principal component analysis. *Chemometrics Intelligent Laboratory Systems*, 1987, **2**(13): 37-52
- Gong Y, Lazebnik S. Iterative quantization: a procrustean approach to learning binary codes. In: *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*. Colorado, USA: IEEE, 2011. 817-824
- Norouzi M, Fleet D. Minimal loss hashing for compact binary codes. In: *Proceedings of the 2011 International Conference on Machine Learning*. Bellevue, Washington, USA: Omnipress, 2011. 353-360
- Liu W, Wang J, Ji R, Jiang Y, Chang S. Supervised hashing with kernels. In: *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*. Providence, Rhode Island, USA: IEEE, 2012. 2074-2081
- Shen F, Shen C, Liu W, Shen H. Supervised discrete hashing. In: *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*. Boston, USA: IEEE, 2015. 37-45
- Alex K, Ilya S, Geoffrey E. ImageNet Classification with deep convolutional neural networks. In: *Proceedings of the 25th International Conference on Neural Information Processing*. USA: Curran Associates Inc, 2012. 1097-1105

- 13 Zhao F, Huang Y, Wang L, Tan T. Deep semantic ranking based hashing for multi-label image retrieval. In: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston, MA, USA: IEEE, 2015. 1556–1564
- 14 Luo Jian-Hao, Wu Jian-Xin. A survey on fine-grained image categorization using deep convolutional features. *Acta Automatica Sinica*, 2017, **43**(8): 1306–1318
(罗建豪, 吴建鑫. 基于深度卷积特征的细粒度图像分类研究综述. 自动化学报, 2017, **43**(8): 1306–1318)
- 15 Zhang Hui, Wang Kun-Feng, Wang Fei-Yue. Advances and perspectives on applications of deep learning invisual object detection. *Acta Automatica Sinica*, 2017, **43**(8): 1289–1305
(张慧, 王坤峰, 王飞跃. 深度学习在目标视觉检测中的应用进展与展望. 自动化学报, 2017, **43**(8): 1289–1305)
- 16 Yin Hong-Peng, Chen Bo, Chai Yi, Liu Zhao-Dong. Vision-based object detection and tracking: a review. *Acta Automatica Sinica*, 2016, **42**(10): 1466–1489
(尹宏鹏, 陈波, 柴毅, 刘兆栋. 基于视觉的目标检测与跟踪综述. 自动化学报, 2016, **42**(10): 1466–1489)
- 17 Xia R, Pan Y, Lai H, Liu C, Yan S. Supervised hashing for image retrieval via image representation learning. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence. CA: AAAI, 2014. 2156–2162
- 18 Liu H, Wang R, Shan S, Chen X. Deep supervised hashing for fast image retrieval. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. New York, USA: IEEE, 2016. 2064–2072
- 19 Li W, Wang S, Kang W. Feature learning based deep supervised hashing with pairwise labels. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence. New York, USA: AAAI Press, 2016. 1711–1717
- 20 Cao Z, Long M, Wang J, Philip S. HashNet: deep learning to hash by continuation. In: Proceedings of the 2017 International Conference on Computer Vision. Venice, Italy: IEEE, 2017. 5609–5618
- 21 Zhang P, Zhang W, Li W, Guo M. Supervised hashing with latent factor models. In: Proceedings of the 37th international ACM SIGIR conference on Research and development in information retrieval. New York, NY, USA: ACM, 2014. 173–182
- 22 Kulis B, Darrell T. Learning to hash with binary reconstructive embeddings. In: Proceedings of the 2009 Advances in Neural Information Processing Systems. USA: Curran Associates Inc, 2009, 22: 1042–1050
- 23 Liu Ye, Pan Yan, Xia Rong-Kai, Liu Di, Yin Jian. FP-CNNH: A fast image hashing algorithm based on deep convolutional neural network. *Computer Science*, 2016, **43**(9): 39–46
(刘冶, 潘炎, 夏榕楷, 刘荻, 印鉴. FP-CNNH: 一种基于深度卷积神经网络的快速图像哈希算法. 计算机科学, 2016, **43**(9): 39–46)
- 24 Lai H, Pan Y, Liu Y, Yan S. Simultaneous feature learning and hash coding with deep neural networks. In: Proceedings of the 2015 International Conference on Computer Vision and Pattern Recognition. Boston, MA, USA: IEEE, 2015. 3270–3278
- 25 Zhu H, Long M, Wang J, Cao Y. Deep hashing network for efficient similarity retrieval. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence. Phoenix, Arizona: AAAI Press, 2016. 2415–2421
- 26 Abouelnaga Y, Ali O, Rady H, Moustafa M. CIFAR-10: KNN-Based ensemble of classifiers. In: Proceedings of the 2016 International Conference on Computational Science and Computational Intelligence. Las Vegas, NV, USA: IEEE, 2016. 1192–1195
- 27 Chua T, Tang J, Hong R, Li H, Luo Z, Zheng Y. NUS-WIDE: A real-world web image database from National University of Singapore. In: Proceedings of the 2009 ACM International Conference on Image and Video Retrieval. New York, NY, USA: ACM, 2009. 368–375
- 28 Yao Tao, Kong Xiang-Wei, Fu Hai-Yan, Tian Qi. Projective dictionary learning hashing for cross-modal retrieval. *Acta Automatica Sinica*, 2018, **44**(8): 1475–1485
(姚涛, 孔祥维, 付海燕, 田齐. 基于映射字典学习的跨模态哈希检索. 自动化学报, 2018, **44**(8): 1475–1485)



汪海龙 北京工业大学信息学部硕士研究生。主要研究方向为图像处理与机器学习。E-mail: 18810815820@163.com
(**WANG Hai-Long** Master student at the Faculty of Information Technology, Beijing University of Technology. His research interest covers image processing and machine learning.)



禹晶 北京工业大学信息学部副教授。2011年获清华大学电子工程系博士学位。主要研究方向为图像处理与模式识别。E-mail: jing.yu@bjut.edu.cn
(**YU Jing** Associate professor at the Faculty of Information Technology, Beijing University of Technology. She received her Ph. D. degree from Tsinghua University in 2011. Her research interest covers image processing and pattern recognition.)



肖创柏 北京工业大学信息学部教授。主要研究方向为数字信号处理, 音视频信号处理与网络通信。本文通信作者。E-mail: cbxiao@bjut.edu.cn
(**XIAO Chuang-Bai** Professor at the Faculty of Information Technology, Beijing University of Technology. His research interest covers digital signal processing, audio and video signal processing, and network communication. Corresponding author of this paper.)