计算机视觉课程报告

卢涛
21821236

推荐理由

这次课程报告我要介绍的是CVPR2018 的一篇论文，题目为 Deep Cauchy Hashing for Hamming Space Retrieval[1]. 之所以推荐这篇论文，是因为我的研究方向是哈希技术，最近也是在研究哈希技术在图像检索上的运用。哈希算法由于其计算效率和检索质量，已被广泛应用于近似最近邻搜索的大规模图像检索，而深度哈希通过端到端表示学习和哈希编码进一步提高了检索质量。通过紧凑的散列码，海明空间检索可实现最有效的恒定时间搜索，该搜索可通过哈希表查找而不是线性扫描将给定海明半径内的数据点返回给每个查询。然而，由于错误指定的损失函数，将相关图像集中在小海明球内的能力较弱，现有的深度哈希方法可能会因海明空间检索而表现不佳。

文章特色

这篇文章提出了Deep Cauchy Hashing（DCH），这是一种新颖的深度哈希模型，可生成紧凑且集中的二进制哈希代码，从而实现高效的海明空间检索。它的创新点有两个方面，一个方面是它采用柯西分布函数代替了传统的sigmoid函数来定义概率函数，另一个方面是它提出了一种新的控制量化损失的损失函数。在基于柯西分布的成对交叉熵损失函数和量化损失函数的共同作用下，该模型在图像检索上具有较高的准确率。为了验证模型的有效性，我复现了论文代码，并在NUS-WIDE, CIFAR-10, 和 MS-COCO三个数据集上做了实验，实验结果表明，DCH要优于目前最先进的图像哈希检索技术。

1. Introduction

在大数据时代，大规模和高维度的媒体数据已经在搜索引擎和社交网络中普及。为了保证检索质量和计算效率，近似最近邻（ANN）搜索引起了越来越多的关注。与传统的使用剪枝来建立索引的方法[2]不同，另一个解决方案是将数据进行压缩的哈希方法[3]，它将高维媒体数据用紧凑的二进制代码来表示，并且对于相似的数据生成相似的二进制码。这篇论文重点介绍哈希学习方法[3]，该方法能实现高效的图像检索，并且比独立于数据的哈希方法[4]具有更好的性能。

最近，深度学习哈希方法[5,6,7,8,9,10,11,12]已经表明，深度神经网络可以实现端到端的特征学习，并且具有非线性哈希函数的哈希编码学习能力。这些深度学习哈希方法已经展示了最先进的图像检索性能。特别是，它们证明了使用深度学习的方法学出的哈希编码，不仅能保持原有图片的相似性，还能很好的控制量化损失[9,10,11,12]。

但是，大多数已有的方法旨在最大化哈希码的线性扫描的检索性能。由于线性扫描的时间开销是非常昂贵的，所以这些方法在某种程度上偏离了哈希原始意图，也就是在可接受的准确度下加速检索效率。因此，我们现在应该转向在海明空间上的检索方法[13]，它可以实现在常数时间内进行检索。然而，现有的哈希方法通常由于损失函数设计的不佳导致缺乏将相关图像集中在小海明球内的能力，因此在海明空间上检索方面表现不佳。

这篇文章提出了Deep Cauchy Hashing（DCH），这是一种新颖的深度哈希模型，可生成紧凑且集中的二进制哈希代码，从而实现高效的海明空间检索。其主要思想是设计一个基于柯西分布的成对交叉熵损失，在海明距离大于给定的海明半径阈值的情况下，对相似图像对产生显著的惩罚。同时，作者还进一步提出了基于Cauchy分布的量化损失，这使得该模型能够学习几乎无损的哈希码。两种损失函数都可以在贝叶斯学习框架中导出，并且对海明空间检索有很好的指导作用。该深度模型可以通过反向传播进行端到端的训练学习。实验结果表明DCH可以生成高度集中和紧凑的哈希码，并在三个数据集NUS-WIDE，CIFAR-10 和MS-COCO上拥有最先进的图像检索性能。

2. Deep Cauchy Hashing

给定N个样本的训练集{Xi},其中一些样本对Xi，Xj提供了相似标签Sij，Xi，Xj相似时Sij=1，不相似则Sij=0。Deep hashing需要学习一个非线性哈希函数:

$$f: x \rightarrow h \in \{-1,1\}^k$$

同时保持样本之间的相似性关系。

本文介绍了一种新的Deep Cauchy Hashing（DCH）在端到端框架中实现高效的汉明空间检索，如图 1 所示。

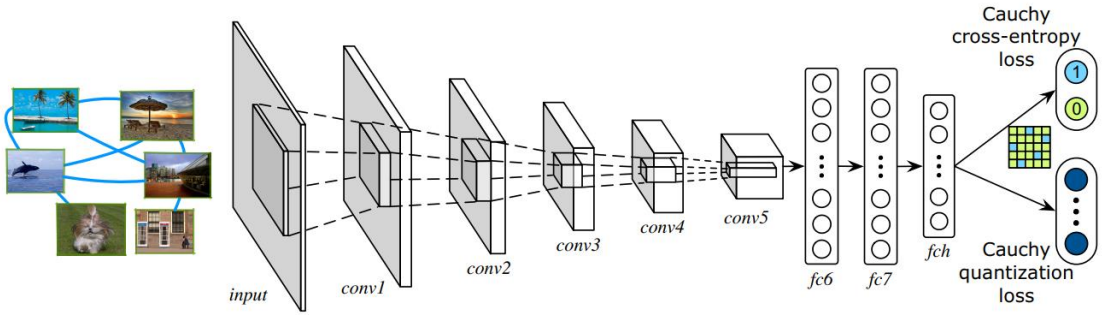


图 1:DCH深度架构图

它接受样本对{(xi,xj,sij)}，进入一个由四部分构成的端对端的 Pipeline：1) CNN 特征提取器; 2) 全连接哈希层（fch）将特征变换成 k 位 哈希码; 3) 柯西交叉熵损失函数; 4) 柯西量化损失函数。

2.1. 海明距离检索

在二进制空间中，我们通常以海明距离来衡量两个哈希码的相似关系。具体来说，就是有多少个相应比特位上的值不一致，海明距离就是几。在检索领域，我们通常所说的最近邻就是海明距离小于等于r ，r 称为海明半径。对于k位的哈希码，会有$N(K,r)=\sum_{k=0}^{r}\binom{K}{k}$个最近邻。虽然每个哈希码的查询时间是O(1)，但是随着k的增大，$N(K,r)$也在迅速增大，所以为了时间效率的考虑，在实际工程中，一般只设r<=2 。因此本文能在r <=2 时极大提升检索效果就显得很有现实意义了。

2.2. 深度模型架构

本文中CNN模型选用的Alexnet[14]，在fc8 后面接了一个fch层，通过h(x)=sgn(x)量化特征，但是在实际优化过程中，sgn函数的梯度不可定义，因此先用Tanh函数将特征数值归一化到[-1,1]区间，再做后续处理。为了进一步保证哈希码的准确性，DCH保留了原始输入的图像对之间的相似性，并且控制哈希码与实数特征向量之间的量化损失。为了实现这一目标，本文提出了两种新型的损失函数：成对柯西交叉熵损失和逐点柯西量化损失，两者都来源于最大后验（MAP）估计框架。

2.3. 贝叶斯学习框架

本节作者运用贝叶斯学习框架来同时优化交叉熵损失与量化损失。给定图像对及其标签$\{(x_i,x_j,s_{ij}):s_{ij}\in\mathcal{S}\}$, 那么哈希码$H=[h_1,h_2,\dots h_n]$的最大后验估计为

$$\log P(\boldsymbol{H}|\mathcal{S}) \propto \log P(\mathcal{S}|\boldsymbol{H})\,P(\boldsymbol{H})$$
$$=\sum_{s_{ij}\in\mathcal{S}} w_{ij}\log P(s_{ij}|\boldsymbol{h}_i,\boldsymbol{h}_j)+\sum_{i=1}^{N}\log P(\boldsymbol{h}_i)$$

其中，P(S|H)是加权似然概率[15]。由于Sij仅可取 0,1 两个数，所以需要用伯努利分布来建模$P(S_{ij}|h_i,h_j)$：

$$P(s_{ij}|\boldsymbol{h}_i,\boldsymbol{h}_j)=\begin{cases}\sigma(\mathrm{d}(\boldsymbol{h}_i,\boldsymbol{h}_j)), & s_{ij}=1\\ 1-\sigma(\mathrm{d}(\boldsymbol{h}_i,\boldsymbol{h}_j)), & s_{ij}=0\end{cases}$$
$$=\sigma(\mathrm{d}(\boldsymbol{h}_i,\boldsymbol{h}_j))^{s_{ij}}(1-\sigma(\mathrm{d}(\boldsymbol{h}_i,\boldsymbol{h}_j)))^{1-s_{ij}}$$

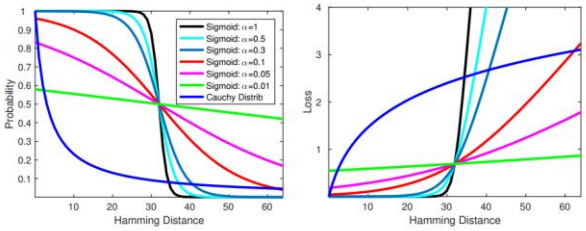
上式中$\mathrm{d}(\mathrm{h_i},\mathrm{h_j})$表示哈希码$\mathrm{h_i},\mathrm{h_j}$之间的海明距离，σ是定义海明距离上的一个分布函数。上式很像逻辑回归中的对数似然概率的表达形式。当$d(h_i,h_j)$很小时，$\mathrm{P}(\mathrm{S_{ij}}|\mathrm{h_i},\mathrm{h_j})$就接近 1，意味着两张图片大概率相似，当$d(h_i,h_j)$很大时，$P(S_{ij}|h_i,h_j)$就接近 0，意味着两张图片不相似。因此，这个式子是是二元类逻辑到成对分类场景的合理扩展，是一个二进制相似性标签的自然解决方案。

2.4. 基于柯西分布的哈希码学习

这是本文最关键的部分，在之前的深度哈希框架中，比如DHN[9], HashNet[12], 都是采用sigmoid函数

$$\sigma(x)=\frac{1}{1+e^{-\alpha x}}$$

来定义概率函数，但实际上这个函数有一个非常大的缺陷，直接看下图：



(a) Probability (b) Loss

在sigmoid函数中，饱和区间过长，即使x>2，函数 σ

(x)也能得到很高的概率，就是说当两个哈希码 的海明距离>2，它们相似的概率依然很高，这样得到的哈希码的判别性就比较差。

针对这一现象，作者提出了一种基于柯西分布的新的 $\sigma(x)$ 形式：

$$\sigma\left(\mathrm{d}\left(\boldsymbol{h}_i, \boldsymbol{h}_j\right)\right)=\frac{\gamma}{\gamma+\mathrm{d}\left(\boldsymbol{h}_i, \boldsymbol{h}_j\right)},$$

从函数图像可以看到，这个函数有一个急剧下降区，使得学到的哈希码具有强的多的判别性。γ 是一个常数因子，可以通过交叉验证来调参。

下面来看MAP中的P(H)这一项，它是哈希码的先验分布。既然我们想学到的是比特编码，h的绝对值就应该靠近 1,所以就可以加上这样一个先验：

$$P\left(\boldsymbol{h}_i\right)=\frac{\gamma}{\gamma+\mathrm{d}\left(\left|\boldsymbol{h}_i\right|, \mathbf{1}\right)},$$

这里同样用到了柯西分布的形式让$|h_i|$更加趋向 1。这个式子就是用来控制量化损失的。

上面所有的公式中，d一直用的是海明距离计算函数，但是在真实的后向算法中，我们需要一种实数松弛来实现可导的梯度。这里作者运用到了一个海明距离的的性质：

$$\begin{aligned}\mathrm{d}\left(\boldsymbol{h}_i, \boldsymbol{h}_j\right) &= \frac{K}{4}\left\|\frac{\boldsymbol{h}_i}{\left\|\boldsymbol{h}_i\right\|}-\frac{\boldsymbol{h}_j}{\left\|\boldsymbol{h}_j\right\|}\right\|_2^2 \\ &= \frac{K}{2}\left(1-\cos\left(\boldsymbol{h}_i, \boldsymbol{h}_j\right)\right).\end{aligned}$$

作者直接将fch层可得的连续型数值特征带入上式来实现实数松弛。于是将上面各式带入MAP的式子中，对其取反，就得到了最终的损失函数：

$$\min_{\Theta} L+\lambda Q,$$

$$L=\sum_{s_{ij}\in\mathcal{S}} w_{ij}\left(s_{ij}\log\frac{\mathrm{d}\left(\boldsymbol{h}_i, \boldsymbol{h}_j\right)}{\gamma}+\log\left(1+\frac{\gamma}{\mathrm{d}\left(\boldsymbol{h}_i, \boldsymbol{h}_j\right)}\right)\right),$$

$$Q=\sum_{i=1}^{N}\log\left(1+\frac{\mathrm{d}\left(\left|\boldsymbol{h}_i\right|, \mathbf{1}\right)}{\gamma}\right),$$

其中L是柯西交叉熵损失函数，用于保持原有图片对的相似性，Q是柯西量化损失，用来控制哈希码与实数向量之间的量化损失，λ 是调节两个损失函数的权重因子。

基于MAP估计，DCH可以通过联合保持成对相似性并控制量化误差来实现紧凑哈希码的统计学最优学习。最后，通过简单的符号函数sgn来获得K位二进制哈希码。值得注意的是，由于我们已经最小化了量化误差，因此最终的检索质量损失非常小，所以图像检索的性能非常高。

3. Experiments

本文原作者的实验是用tensorflow实现的，代码地址 https://github.com/thulab/DeepHash. 我用pytorch重新复现了论文代码，代码地址：https://github.com/3140102441/DCH--pytorch

我在NUS-WIDE[16], CIFAR-10[9], 和 MS-COCO[17]三个数据集上做了实验，使用map作为验证指标，结果如下：

3.1. 与其他方法对比

图 2 是DCH与已有的哈希检索方法的map值对比，可以看到，当海明半径小于 2 时，DCH均取得了最好的效果，并且优势还相当可观。从中，我们可以得出以下几个结论：

1）DCH要比在浅层模型中表现最好的SDH[7]的map值平均高出 15%左右，这说明浅层模型无法通过端对端的方式，学到深层的图片特征以及哈希码，这也是深层模型要比浅层模型好的原因。

2) 深度哈希方法DHN[9]和HashNet要比CNNH[5]和DNNH[6]好，原因在于前两者不光保持了原有图片的相似性，而且还控制了量化损失。

3）DCH要比原有的方法中表现最好的HashNet要好，原因有两方面，一方面是DCH在基于柯西分布的基础上保持图片的相似性，这导致其结果在海明半径小于 2 的时候，效果比较好。另一方面，DCH的交叉熵损失和量化损失都是经过归一化的，这也导致了在海明空间中好的表现。

3.2. 交叉熵损失和量化损失的作用

从图 3 可以看到，DCH在没有交叉熵损失和量化损失的情况下，检索的性能都出现了下滑，其中，在没有交叉熵损失的情况下，性能下降了 5%左右，而在没有量化损失的情况下，结果下降了 3%左右，这说明了两个损失函数对效果提升都有帮助，前者提升更大。

3.3. 可视化

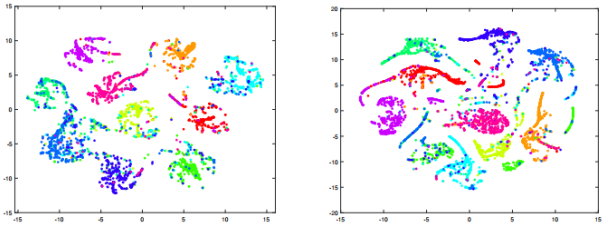
图 4 是DCH与HashNet在cifar10 数据集上学到的哈希码的t-SNE[18]可视化，可以看到，DCH生成的哈希码有着清晰的结构界限，每个不同的图片类别都被很好地区分开来，而HashNet的类别区别就没有那么明显，由此可验证了，使用柯西分布来代替sigmoid分布，可以帮助模型学到更加结构化明显的哈希码，从而实现更加有效的图像检索效果。

Method	NUS-WIDE				MS-COCO				CIFAR-10			
	16 bits	32 bits	48 bits	64 bits	16 bits	32 bits	48 bits	64 bits	16 bits	32 bits	48 bits	64 bits
ITQ-CCA [12]	0.5706	0.4397	0.0825	0.0051	0.5949	0.5612	0.0585	0.0105	0.4258	0.4652	0.4774	0.4932
BRE [16]	0.5502	0.5422	0.4128	0.2202	0.5625	0.5498	0.4214	0.4014	0.4216	0.4519	0.4002	0.3438
KSH [22]	0.5185	0.5659	0.4102	0.0608	0.5797	0.5532	0.2338	0.0216	0.4368	0.4585	0.4012	0.3819
SDH [28]	0.6681	0.6824	0.5979	0.4679	0.6449	0.6766	0.5226	0.5108	0.5620	0.6428	0.6069	0.5012
CNNH [33]	0.5843	0.5989	0.5734	0.5729	0.5602	0.5685	0.5376	0.5058	0.5512	0.5468	0.5454	0.5364
DNNH [17]	0.6191	0.6216	0.5902	0.5626	0.5771	0.6023	0.5235	0.5013	0.5703	0.5985	0.6421	0.6118
DHN [36]	0.6901	0.7021	0.6685	0.5664	0.6749	0.6680	0.5151	0.4186	0.6929	0.6445	0.5835	0.5883
HashNet [4]	0.6944	0.7147	0.6736	0.6190	0.6851	0.6900	0.5589	0.5344	0.7476	0.7776	0.6399	0.6259
DCH	**0.7401**	**0.7720**	**0.7685**	**0.7124**	**0.7010**	**0.7576**	**0.7251**	**0.7013**	**0.7901**	**0.7979**	**0.8071**	**0.7936**

图 2:在不同数据集上DCH与其他算法的比较

Method	NUS-WIDE				MS-COCO				CIFAR-10			
	16 bits	32 bits	48 bits	64 bits	16 bits	32 bits	48 bits	64 bits	16 bits	32 bits	48 bits	64 bits
DCH	**0.7401**	**0.7720**	**0.7685**	**0.7124**	**0.7010**	**0.7576**	**0.7251**	**0.7013**	**0.7901**	**0.7979**	**0.8071**	**0.7936**
DCH-Q	0.7086	0.7458	0.7432	0.7028	0.6785	0.7238	0.6982	0.6913	0.7645	0.7789	0.7858	0.7832
DCH-C	0.6997	0.7205	0.6874	0.6328	0.6767	0.6972	0.5801	0.5536	0.7513	0.7628	0.6819	0.6627
DCH-E	0.7178	0.7511	0.7302	0.6982	0.6826	0.7128	0.6803	0.6735	0.7598	0.7739	0.7495	0.7287

图 3:交叉熵损失和量化损失对DCH算法的影响



(a) DCH (b) HashNet

图 4: 哈希码的t-SNE可视化

图 5:DCH和HashNet的top-10 图片检索结果

References

[1] Deep Cauchy Hashing for Hamming Space Retrieval, Yue Cao, Mingsheng Long, Bin Liu, Jianmin Wang, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018

[2] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2(1):1–19, Feb.2006.

[3] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for large-scale search. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 34(12):2393–2406, 2012.

[4] A. Gionis, P. Indyk, R. Motwani, et al. Similarity search in high dimensions via hashing. In VLDB, volume 99, pages 518–529. ACM, 1999.

[5] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan. Supervised hashing for image retrieval via image representation learning. In AAAI, pages 2156–2162. AAAI, 2014.

[6] H. Lai, Y. Pan, Y. Liu, and S. Yan. Simultaneous feature learning and hash coding with deep neural networks. In CVPR. IEEE, 2015.

[7] F. Shen, C. Shen, W. Liu, and H. Tao Shen. Supervised discrete hashing. In CVPR. IEEE, June 2015.

[8] V. Erin Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou. Deep hashing for compact binary codes learning. In CVPR, pages 2475–2483. IEEE, 2015.

[9] H. Zhu, M. Long, J. Wang, and Y. Cao. Deep hashing network for efficient similarity retrieval. In AAAI. AAAI, 2016.

[10] W.-J. Li, S. Wang, and W.-C. Kang. Feature learning based deep supervised hashing with pairwise labels. In IJCAI, 2016.

[11] H. Liu, R. Wang, S. Shan, and X. Chen. Deep supervised hashing for fast image retrieval. In CVPR, pages 2064–2072, 2016.

[12] Z. Cao, M. Long, J. Wang, and P. S. Yu. Hashnet: Deep learning to hash by continuation. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 5609–5618, 2017.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. CVPR, 2016.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

[15] J. P. Dmochowski, P. Sajda, and L. C. Parra. Maximum likelihood in cost-sensitive learning: Model specification, approximations, and upper bounds. Journal of Machine Learning Research (JMLR), 11(Dec):3313–3332, 2010.

[16] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng. Nus-wide: A real-world web image database from national university of singapore. In ICMR. ACM, 2009.

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Common objects in context. In ECCV, pages 740–755. Springer, 2014.

[18] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. Journal of Machine Learning Research, 9: 2579–2605, Nov 2008.