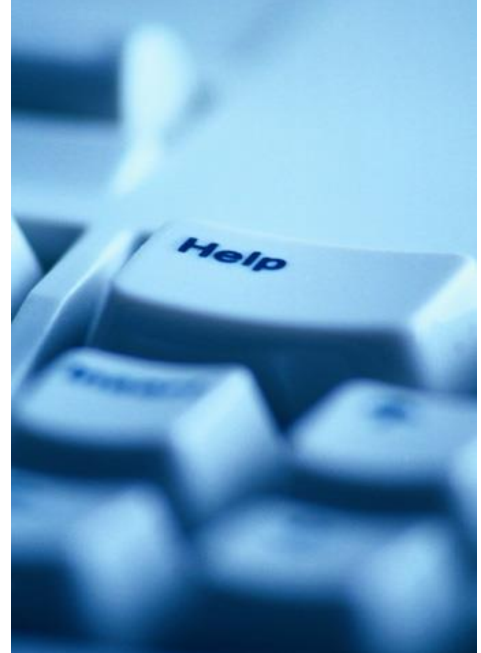
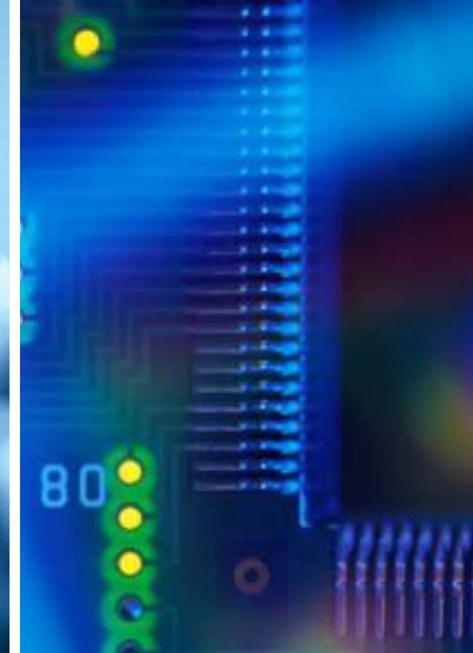




Triplet-Based Hashing Network for Cross-Modal Retrieval

From 2018 TIP

Cheng Deng, *Student Member, IEEE*, Zhaojia Chen, Xianglong Liu, Xinbo Gao, *Senior Member, IEEE*, and Dacheng Tao, *Fellow, IEEE*

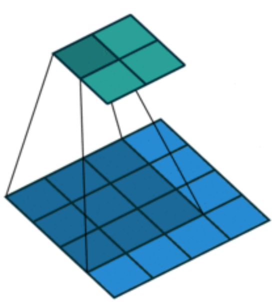
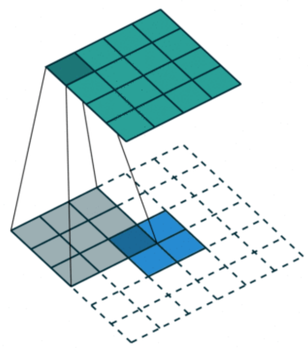
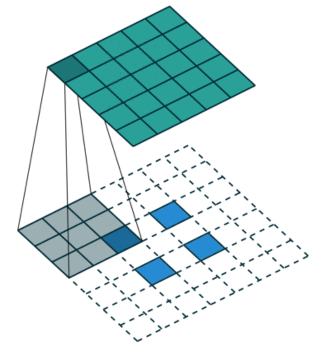
背景介绍

1.为什么使用hash
跨模态分类用于将同一分类下的不同模态进行分类，同时为了方便分类的查询，将不同分类的对象映射到hash code中。Hash code具有低存储耗费和快速查询的优势，所以利用hashing对跨模态数据进行检索。
2.为什么使用深度学习
随着硬件的条件提升，传统方法逐渐被机器学习的方法替代，虽然机器学习的数学基础难以证明，但机器学习的在一些特定的方向，如图像、音频、自然语言处理等，相较于传统方面在精度上有明显提升。而深度学习是机器学习中效果显著的方法。
3.跨模态都包含什么模态
在论文中多数以图像和文本两个模态为例，实际中还包括了视频、图画、素描、空间文本、文字描述等

相关概念及算法

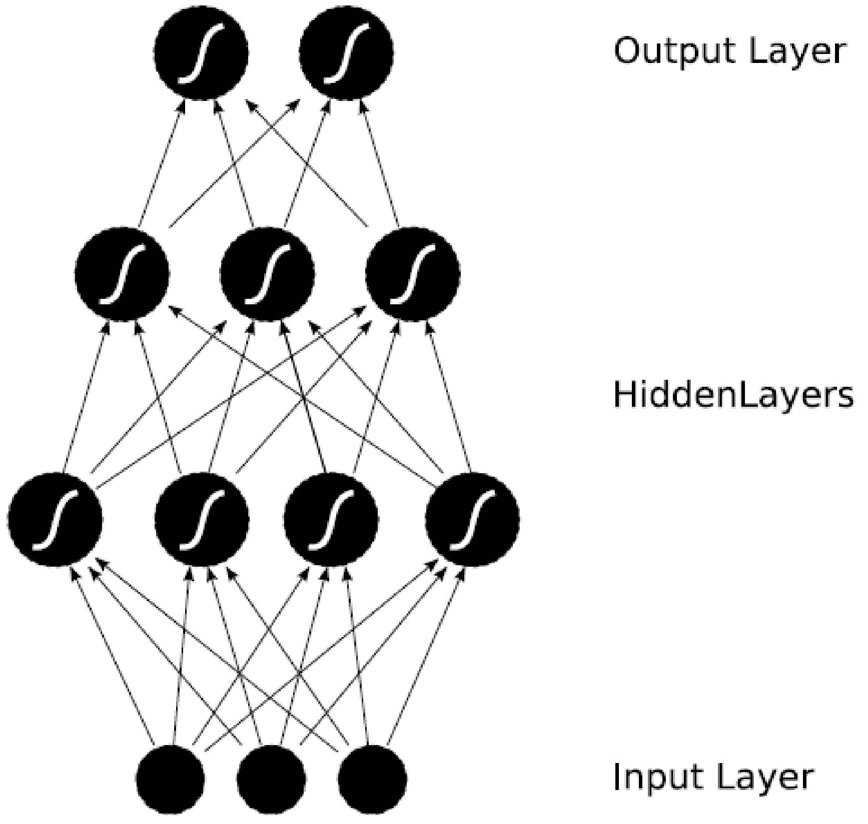
1. CNN

Convolution neural network（卷积神经网络）通过卷积核对输入中指定大小的矩阵进行相应位置的乘法，并将结果加和输出。按照一般理解，卷积其实就是一种滤波，通过卷积滤波后的图像中对应于该卷积核的特征会突出显示，配合卷积之后的一般是pooling（池化）操作，用来将通过卷积后获得的突出特征进行筛选，剔除非强调特征。随着卷积与池化的结合，将一个图像的浅层语义到深层语义依次筛选出。同时卷积包括其不同类型，如转置卷积(反卷积)、微步卷积。如下图分别为卷积操作、反卷积操作、微步卷积操作

相关概念及算法

2. BOW
Bag of words 用于文本分类中，将文本表示成矢量。即对于一个文本，忽略次序、语法、句法仅看作为一个单词的集合(中文也是词组).利用多层感知机来对文本矢量进行特征提取。感知机模型如右图所示。



相关概念及算法

3. Triplet loss
三元组loss最初在人脸识别中起到了很好的表现，即2015年CVPR的FaceNet, triplet loss定义如下：
对于每一对正样本(P)和负样本(N), 我们选择一个接近于正样本的询问对象(A),使得询问对象离正样本的距离小于负样本的距离。

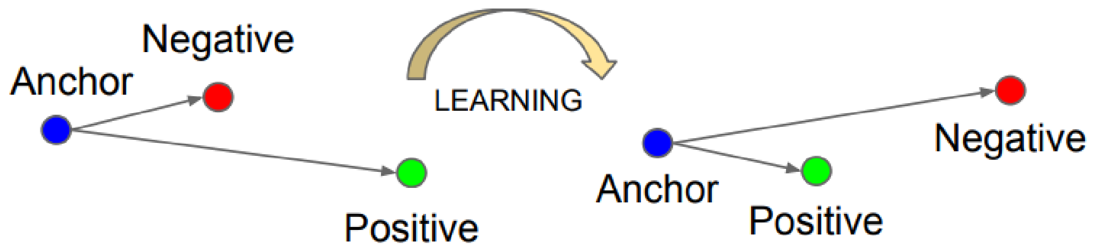


Figure 3. The **Triplet Loss** minimizes the distance between an *anchor* and a *positive*, both of which have the same identity, and maximizes the distance between the *anchor* and a *negative* of a different identity.

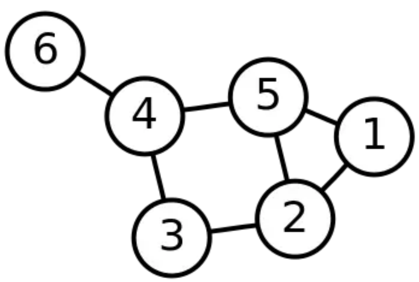
相关概念及算法

4. 相关数学定义

Frobenius 范数:

$$||A||F = \sqrt{\sum i = 1^m \sum_{j=1}^{n} |a_{i,j}|^2} = \sqrt{tr(A^H A)} \text{ 其中 } A^H \text{ 为 } A \text{ 的共轭矩阵}$$

拉普拉斯矩阵： L = D – S (图的度矩阵 – 邻接矩阵)



度矩阵D　　邻接矩阵S　　拉普拉斯矩阵L

$$\begin{pmatrix} 2&0&0&0&0&0\\ 0&3&0&0&0&0\\ 0&0&2&0&0&0\\ 0&0&0&3&0&0\\ 0&0&0&0&3&0\\ 0&0&0&0&0&1 \end{pmatrix} - \begin{pmatrix} 0&1&0&0&1&0\\ 1&0&1&0&1&0\\ 0&1&0&1&0&0\\ 0&0&1&0&1&1\\ 1&1&0&1&0&0\\ 0&0&0&1&0&0 \end{pmatrix} = \begin{pmatrix} 2&-1&0&0&-1&0\\ -1&3&-1&0&-1&0\\ 0&-1&2&-1&0&0\\ 0&0&-1&3&-1&-1\\ -1&-1&0&-1&3&0\\ 0&0&0&-1&0&1 \end{pmatrix}$$

Network structure

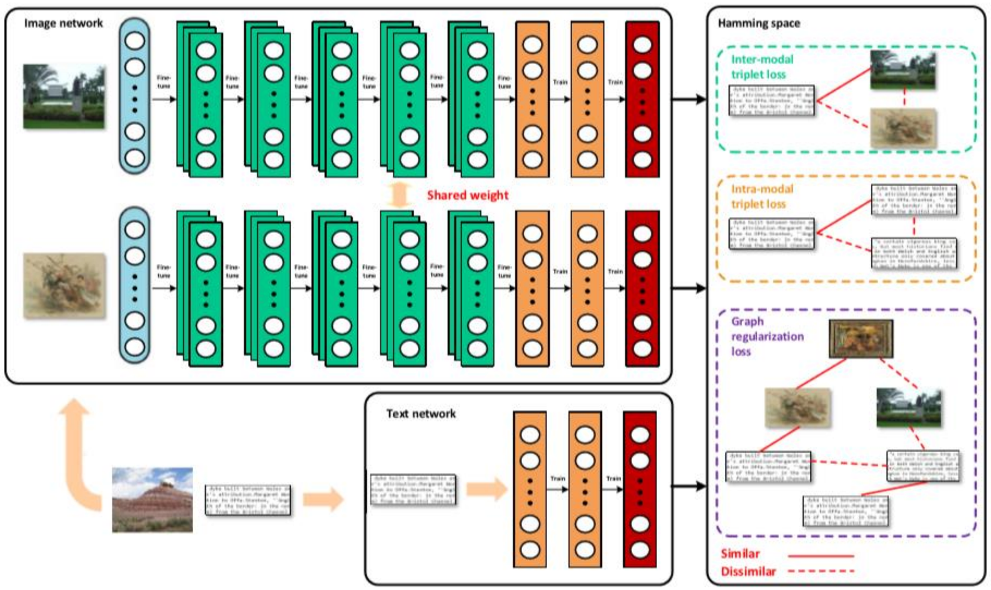


Table 1. Configuration of the CNN for image modality.

Layer	Configuration
conv1	f. 64 × 11 × 11; st. 4 × 4, pad 0, LRN,×2 pool
conv2	f. 265 × 5 × 5; st. 1 × 1, pad 2, LRN,×2 pool
conv3	f. 265 × 3 × 3; st. 1 × 1, pad 1
conv4	f. 265 × 3 × 3; st. 1 × 1, pad 1
conv5	f. 265 × 3 × 3; st. 1 × 1, pad 1,×2 pool
full6	4096
full7	4096
full8	Hash code length c

“LRN” 表示是否使用了 Local Response Normalization 即是否使用正则化处理输出结果，pool为MaxPooling

Table 2. Configuration of the deep neural network for text modality.

Layer	Configuration
full1	Length of BOW vector
full2	4096
full3	Hash code length c

Triplet loss

每次进行实例选取时，会选取3个对象：positive, negative, query，其中query于positive相对于negative更加接近。这样就有了在Face net中于triplet loss相似的本文triplet label likelihood公式：

$$p(\mathcal{T}|\mathbf{F},\mathbf{G},\mathbf{G}) = \prod_{m=1}^{M} p((q_m, p_m, n_m)|\mathbf{F},\mathbf{G},\mathbf{G}). \qquad (1)$$

with

$$p((q_m, p_m, n_m)|\mathbf{F},\mathbf{G},\mathbf{G}) = \sigma(\theta_{q_m^y p_m^x} - \theta_{q_m^y n_m^x} - \alpha), \qquad (2)$$

其中F表示图像提取出的feature map，G表示文本提取出的feature map.

where $\theta_{q_m^y p_m^x} = \frac{1}{2}\mathbf{F}_{*q_m}^{\top}\mathbf{G}_{*p_m}$, $\theta_{q_m^y n_m^x} = \frac{1}{2}\mathbf{F}_{*q_m}^{\top}\mathbf{G}_{*n_m}$, $\sigma(x)$ is the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, and the threshold α is a margin that is enforced between positive and negative pairs, a hyper-parameter. $\mathbf{F}_{*i}^{\top} = f^y(\mathbf{y}_i; w_y)$, and $\mathbf{G}_{*i}^{\top} = f^x(\mathbf{x}_i; w_x)$, where w_x, w_y are the network parameters of textual modality and image modality, respectively.

Inter-modal Triplet loss (J^{inter})

用来区分同一类的不同模态之间的feature map的距离，即用来区分同一类内图像和文本之间的距离。 其中，Image to Text 的 inter-modal triplet loss J1和Text to Image 的 inter-modal triplet loss表示为：

$$\begin{aligned} J_1 &= -log\ p(\mathcal{T}|\mathbf{F},\mathbf{G},\mathbf{G}) \\ &= -\sum_{m=1}^{M} log\ p((q_m,p_m,n_m)|\mathbf{F},\mathbf{G},\mathbf{G}) \\ &= \sum_{m=1}^{M}(\theta_{q_m^y p_m^x} - \theta_{q_m^y n_m^x} - \alpha - log(1+e^{\theta_{q_m^y p_m^x} - \theta_{q_m^y n_m^x} - \alpha})), \quad (3) \end{aligned}$$

$$\begin{aligned} J_2 &= -log\ p(\mathcal{T}|\mathbf{G},\mathbf{F},\mathbf{F}) \\ &= -\sum_{m=1}^{M} log\ p((q_m,p_m,n_m)|\mathbf{G},\mathbf{F},\mathbf{F}) \\ &= -\sum_{m=1}^{M}(\theta_{q_m^x p_m^y} - \theta_{q_m^x n_m^y} - \alpha - log(1+e^{\theta_{q_m^x p_m^y} - \theta_{q_m^x n_m^y} - \alpha})), \quad (4) \end{aligned}$$

这里通过将query对象选取与positive对象同类但不同形态的对象，通过loss的减少来缩短同类型下不同模态对象之间的hash code的hamming distance，同时增加不同类型之间不同模态对象的距离 最后 得到的inter-modal triplet loss = J1 + J 2

Intra-modal Triplet loss (J^{intra})

用来区分同一模态下对象之间的分类距离。 其中对于图像类型的对象区分lossJ3及对于文本类型的对象区分lossJ4采用如下计算方式

$$
\begin{aligned}
J_3 &= -log\ p(\mathcal{T}|\mathbf{F}) \\
&= -\sum_{m=1}^{M} log\ p((q_m, p_m, n_m)|\mathbf{F}) \\
&= -\sum_{m=1}^{M} (\theta_{q_m^y p_m^y} - \theta_{q_m^y n_m^y} - \alpha - log(1 + e^{\theta_{q_m^y p_m^y} - \theta_{q_m^y n_m^y} - \alpha})),
\end{aligned} \tag{6}
$$

$$
\begin{aligned}
J_4 &= -log\ p(\mathcal{T}|\mathbf{G}) \\
&= -\sum_{m=1}^{M} log\ p((q_m, p_m, n_m)|\mathbf{G}) \\
&= -\sum_{m=1}^{M} (\theta_{q_m^x p_m^x} - \theta_{q_m^x n_m^x} - \alpha - log(1 + e^{\theta_{q_m^x p_m^x} - \theta_{q_m^x n_m^x} - \alpha})),
\end{aligned} \tag{7}
$$

graph regularization loss (J^{re})

通过图谱的之间的距离分类，对得到的hash code进行无监督分类

$$J_{re} = \gamma(\|\mathbf{B}^y - \mathbf{F}\|_F^2 + \|\mathbf{B}^x - \mathbf{G}\|_F^2) + \eta(\|\mathbf{F}\cdot\mathbf{1}\|_F^2 + \|\mathbf{G}\cdot\mathbf{1}\|_F^2) + \beta tr(\mathbf{BLB}^\top) \quad s.t.\ \ \mathbf{B} = \mathbf{B}^x = \mathbf{B}^y \in \{-1, 1\}^{k\times N}, \tag{9}$$

where $\mathbf{B}^x$ is the hash codes of textual modality and $\mathbf{B}^y$ is the hash codes of image modality. The first and second terms represent the quantization error to solve the relaxed problem. Simultaneously, the third and fourth terms are used to make the balanced bit such that the number of 1 and -1 for each bit on the hash codes should be nearly the same. γ, η and β are parameters employed to balance the weight of each part.

这里Bx By 分别是从文本模态和图像模态得出的hash code，L为B的拉普拉斯矩阵，γ、η、β为用来平衡的参数，文中并没给出训练时的取值，但给出在验证集上的的取值依次为100，50，1 对于图谱的学习过程，定义如下：

$$\frac{1}{2}\sum_{i,j=1}^{N}\|\mathbf{b}_i - \mathbf{b}_j\|^2\mathbf{S}_{ij} = tr(\mathbf{BLB}^\top),$$

$$S_{i,j}\begin{cases}1 & b_i \text{ 相似于} b_j\\ 0 & otherwise\end{cases}$$

Objective function

结合上述三种loss：不同模态之间的距离，同模态之间的距离，得出hash code的聚类距离 于是得到我们的目标函数

$$\min_{\mathbf{B},w_x,w_y} J = \min_{\mathbf{B},w_x,w_y} J_{inter} + J_{intra} + J_{re}. \tag{10}$$

Training

对于三个目标的优化，本文采用固定两个，优化一个的方式。训练方式如右图所示

这里标出了query对象的选取方式，对于每个batch，先对CNN-F进行训练，然后训练MPL模型

Input:
Text set $\mathbf{X}$, image set $\mathbf{Y}$, and the set of triplet labels $\mathcal{T}$.
Output:
Parameters w_x and w_y of the deep neural networks, and binary code matrix $\mathbf{B}$.
Initialization
Initialize neural parameters w_X and w_y, mini-batch size $N_x = N_y = 128$, and iteration number $t_x = N/N_x$, $t_y = N/N_y$.
repeat
Update $\mathbf{B}$ according to (12).
for iter=$1, 2, \cdots, t_x$ **do**
Randomly sample N_x instances from $\mathbf{X}$ to construct a mini-batch $\mathbf{X}_{N_x}$ and make up a triplet set where the query instances come from $\mathbf{X}_{N_x}$.
For each sampled instances $\mathbf{x}_i$ in the mini-batch, calculate $\mathbf{G}_{*i} = f(\mathbf{x}_i; w_x)$ by forward propagation.
Calculate the derivative according to (13).
Update parameter w_x using back propagation.
end for
for iter=$1, 2, \cdots, t_y$ **do**
Randomly sample N_y instances from $\mathbf{Y}$ to construct a mini-batch $\mathbf{Y}_{N_y}$ and make up a triplet set where the query instances come from $\mathbf{Y}_{N_y}$.
For each sampled instances $\mathbf{y}_i$ in the mini-batch, calculate $\mathbf{F}_{*i} = f(\mathbf{y}_i; w_y)$ by forward propagation.
Calculate the derivative according to (14).
Update parameter w_y using back propagation.
end for
until a fixed number of iterations;

Training

updating B

当CNN-F和MPL的参数确定后，就可以用来更新B的输出，此时loss只有graph regularization loss在起作用，这里sign(x) = 1 if x >= 0 else 0
identity matrix 为 单位矩阵

When w_x and w_y are fixed, the objective function in (10) can be expanded as follows:

$$\min_{\mathbf{B}} \ \gamma tr(\mathbf{B}^\top \mathbf{B} - \mathbf{F}\mathbf{B}^\top - \mathbf{G}\mathbf{B}^\top) + \beta tr(\mathbf{B}\mathbf{L}\mathbf{B}^\top)$$
$$s.t.\ \mathbf{B} \in \{-1, 1\}^{k\times N}. \tag{11}$$

We compute the derivation of (11) with respect to $\mathbf{B}$ and infer that $\mathbf{B}$ should be defined as follows:

$$\mathbf{B} = sign((\mathbf{F}+\mathbf{G})(2\mathbf{I} + \frac{\beta}{\gamma}\mathbf{L})^{-1}), \tag{12}$$

where $\mathbf{I}$ denotes the identity matrix.

Training

updating Wx

当B确定的时候，我们按照训练过程，我们首先更新Wx的值，通过SGD优化器进行BP优化参数。

$$
\begin{aligned}
\frac{\partial J}{\partial \mathbf{G}_{*i}} &= \frac{\partial J_{inter}}{\partial \mathbf{G}_{*i}} + \frac{\partial J_{intra}}{\partial \mathbf{G}_{*i}} + \frac{\partial J_{re}}{\partial \mathbf{G}_{*i}} \\
&= -\frac{1}{2} \sum_{m:(i,p_m,n_m)}^{M} \left(1-\sigma\left(\theta_{ip_m^y} - \theta_{in_m^y} - \alpha\right)\right)\left(\mathbf{F}_{*p_m} - \mathbf{F}_{*n_m}\right) \\
&\quad - \frac{1}{2} \sum_{m:(i,p_m,n_m)}^{M} \left(1-\sigma\left(\theta_{ip_m^x} - \theta_{in_m^x} - \alpha\right)\right)\left(\mathbf{G}_{*p_m} - \mathbf{G}_{*n_m}\right) \\
&\quad + 2\gamma\left(\mathbf{G} - \mathbf{B}\right) + 2\eta \mathbf{G}\mathbf{1}. \qquad (13)
\end{aligned}
$$

Training

updating Wy

这里和上述类似，公式如下

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{F}_{*i}} &= \frac{\partial J_{inter}}{\partial \mathbf{F}_{*i}} + \frac{\partial J_{intra}}{\partial \mathbf{F}_{*i}} + \frac{\partial J_{re}}{\partial \mathbf{F}_{*i}} \\ &= -\frac{1}{2}\sum_{m:(i,p_m,n_m)}^{M}\left(1-\sigma\left(\theta_{ip_m^x}-\theta_{in_m^x}-\alpha\right)\right)\left(\mathbf{G}_{*p_m}-\mathbf{G}_{*n_m}\right) \\ &\quad -\frac{1}{2}\sum_{m;(i,p_m,n_m)}^{M}\left(1-\sigma\left(\theta_{ip_m^y}-\theta_{in_m^y}-\alpha\right)\right)\left(\mathbf{F}_{*p_m}-\mathbf{F}_{*n_m}\right) \\ &\quad +2\gamma\left(\mathbf{F}-\mathbf{B}\right)+2\eta\mathbf{F}\mathbf{1}. \end{aligned} \tag{14}$$

Triplet sample

对于每次迭代如何选取query对象，这里给出了明确说明。 在每次随机选取P个anchor，然后随机选取M1个正样本和M2个负样本，保证anchor到正样本的距离比到负样本距离短。这样就获得了P * M1 * M2个triplet sample

Experiments

- Datasets:
  (1) MIRFlickr25k
  (2) NUS-WIDE
- Evaluation Criteria:
  (1) mAP
  (2) Precision-Recall curve
  (3) TopN-precision curve
- control experiment（对照试验）

- Baselines:
  (1) CMFH
  (2) SCM
  (3) LSSH
  (4) STMH
  (5) CVH
  (6) SePH
  (7) DCMH
  (8) PRDH

Deep learning

control experiment

本文利用对照试验，测试了三组loss的作用效果，测试结果以mAP为准

TABLE VII

COMPARISON OF DIFFERENT LOSS FUNCTIONS IN TERMS OF MAP. BEST ACCURACY IS SHOWN IN BOLDFACE. THE CODE LENGTH IS 16

Dataset/Loss		$J_{intra} + J_{inter}$	$J_{inter} + J_{re}$	$J_{intra} + J_{re}$	$J_{intra} + J_{inter} + J_{re}$
MIRFlickr25k	$I \rightarrow T$	0.7104	0.6670	0.5800	**0.7110**
	$T \rightarrow I$	0.7414	0.6830	0.5938	**0.7422**
NUSWIDE	$I \rightarrow T$	0.6245	0.5787	0.3750	**0.6393**
	$T \rightarrow I$	0.6597	0.6050	0.4058	**0.6647**

证明了三个loss之间的关系密不可分，缺一不可

Performance of MAP

MIRFLICKR25K

TABLE III

Comparison to Baselines With Hand-Crafted Features on MIRFLICKR-25K in Terms of MAP. Best Accuracy Is Shown in Boldface

Task/MIRFlickr25k	Methods	Code Length		
		16 bits	32 bits	64 bits
Image Query v.s. Text Database	CMFH [20]	0.5804	0.5790	0.5797
	SCM [33]	0.6153	0.6279	0.6288
	LSSH [21]	0.5784	0.5804	0.5797
	STMH [39]	0.5876	0.5951	0.5942
	CVH [32]	0.6067	0.6177	0.6157
	SePH [23]	0.6441	0.6492	0.6508
	DCMH [24]	0.7056	0.7035	0.7140
	PRDH [25]	0.6819	0.6917	0.6913
	TDH	**0.7110**	**0.7228**	**0.7289**
Text Query v.s. Image Database	CMFH [20]	0.5728	0.5778	0.5779
	SCM [33]	0.6102	0.6184	0.6192
	LSSH [21]	0.5898	0.5927	0.5932
	STMH [39]	0.5763	0.5877	0.5826
	CVH [32]	0.6026	0.6041	0.6017
	SePH [23]	0.6455	0.6474	0.6506
	DCMH [24]	0.7311	0.7487	0.7499
	PRDH [25]	0.7340	0.7397	0.7418
	TDH	**0.7422**	**0.7500**	**0.7548**

TABLE V

Comparison to Baselines With CNN-F Features on MIRFLICKR-25K in Terms of MAP. Best Accuracy Is Shown in Boldface

Task/MIRFlickr25k	Methods	Code Length		
		16 bits	32 bits	64 bits
Image Query v.s. Text Database	CMFH [20]	0.5451	0.5455	0.5451
	SCM [33]	0.6095	0.6139	0.6143
	LSSH [21]	0.5712	0.5822	0.5880
	STMH [39]	0.5944	0.5948	0.6047
	CVH [32]	0.5378	0.5378	0.5378
	SePH [23]	0.6984	0.7048	0.7086
	DCMH [24]	0.7056	0.7035	0.7140
	PRDH [25]	0.6819	0.6917	0.6913
	TDH	**0.7110**	**0.7228**	**0.7289**
Text Query v.s. Image Database	CMFH [20]	0.5354	0.5353	0.5352
	SCM [33]	0.6316	0.6349	0.6360
	LSSH [21]	0.5687	0.5707	0.5689
	STMH [39]	0.5915	0.5931	0.6084
	CVH [32]	0.5399	0.5352	0.5412
	SePH [23]	0.6438	0.6460	0.6518
	DCMH [24]	0.7311	0.7487	0.7499
	PRDH [25]	0.7340	0.7397	0.7418
	TDH	**0.7422**	**0.7500**	**0.7548**

Performance of MAP

NUSWIDE

TABLE IV

COMPARISON TO BASELINES WITH HAND-CRAFTED FEATURES ON NUS-WIDE IN TERMS OF MAP. BEST ACCURACY IS SHOWN IN BOLDFACE

Task/NUS-WIDE	Methods	Code Length		
		16bits	32bits	64bits
Image Query v.s. Text Database	CMFH [20]	0.3825	0.3858	0.3890
	SCM [33]	0.4904	0.4945	0.4992
	LSSH [21]	0.3900	0.3924	0.3962
	STMH [39]	0.4344	0.4461	0.4534
	CVH [32]	0.3687	0.4182	0.4602
	SePH [23]	0.5314	0.5340	0.5429
	DCMH [24]	0.6141	0.6167	0.6427
	PRDH [25]	0.5874	0.6154	0.6232
	TDH	**0.6393**	**0.6626**	**0.6754**
Text Query v.s. Image Database	CMFH [20]	0.3915	0.3944	0.3990
	SCM [33]	0.4595	0.4650	0.4691
	LSSH [21]	0.4286	0.4248	0.4248
	STMH [39]	0.3845	0.4089	0.4181
	CVH [32]	0.3646	0.4024	0.4339
	SePH [23]	0.5086	0.5055	0.5710
	DCMH [24]	0.6591	0.6487	**0.6847**
	PRDH [25]	0.6303	0.6432	0.6568
	TDH	**0.6647**	**0.6758**	0.6803

TABLE VI

COMPARISON TO BASELINES WITH CNN-F FEATURES ON NUS-WIDE IN TERMS OF MAP. BEST ACCURACY IS SHOWN IN BOLDFACE

Task/NUS-WIDE	Methods	Code Length		
		16bits	32bits	64bits
Image Query v.s. Text Database	CMFH [20]	0.3552	0.3549	0.3545
	SCM [33]	0.4561	0.4664	0.4697
	LSSH [21]	0.4425	0.4457	0.4539
	STMH [39]	0.5269	0.5210	0.5461
	CVH [32]	0.3671	0.3671	0.3672
	SePH [23]	0.6224	0.6469	0.6609
	DCMH [24]	0.6141	0.6167	0.6427
	PRDH [25]	0.5874	0.6154	0.6232
	TDH	**0.6393**	**0.6626**	**0.6754**
Text Query v.s. Image Database	CMFH [20]	0.3724	0.3723	0.3722
	SCM [33]	0.4561	0.4707	0.4799
	LSSH [21]	0.4153	0.4295	0.4415
	STMH [39]	0.5089	0.5160	0.5420
	CVH [32]	0.3642	0.3596	0.3568
	SePH [23]	0.5658	0.5596	0.6016
	DCMH [24]	0.6591	0.6487	**0.6847**
	PRDH [25]	0.6303	0.6432	0.6568
	TDH	**0.6647**	**0.6758**	0.6803

Performance of Precision-Recall Curve

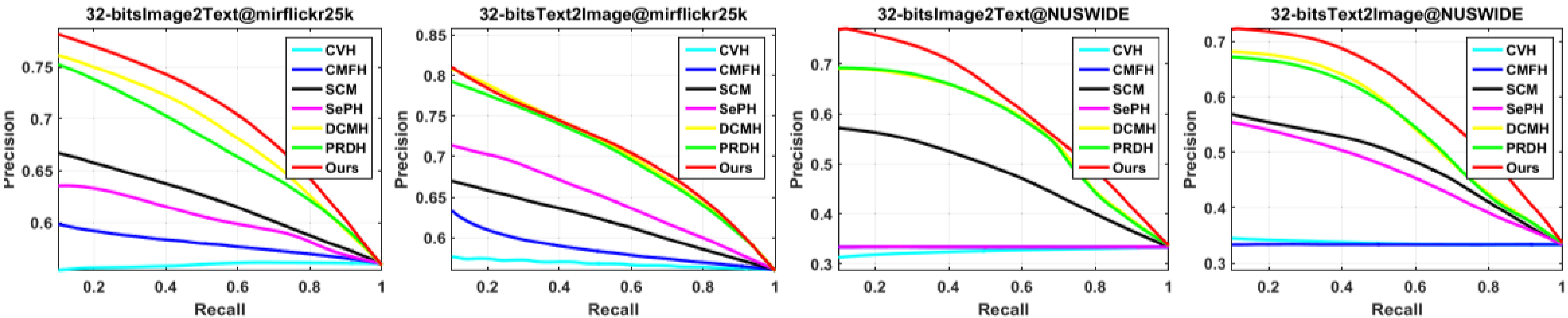


Fig. 2. Precision-recall curves. The baselines are based on hand-crafted features. The code length is 32.

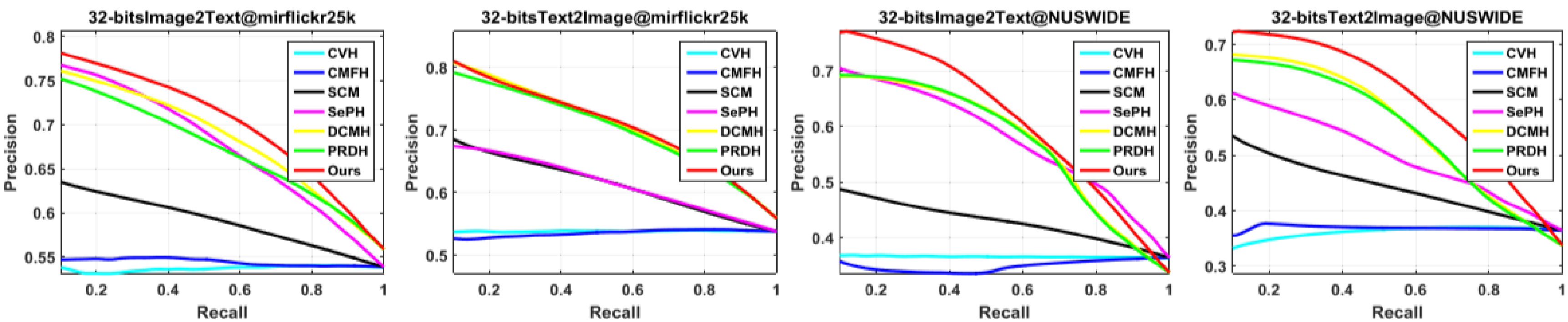


Fig. 4. Precision-recall curves. The baselines are based on CNN-F features. The code length is 32.

Performance of TopN-Precision Curve

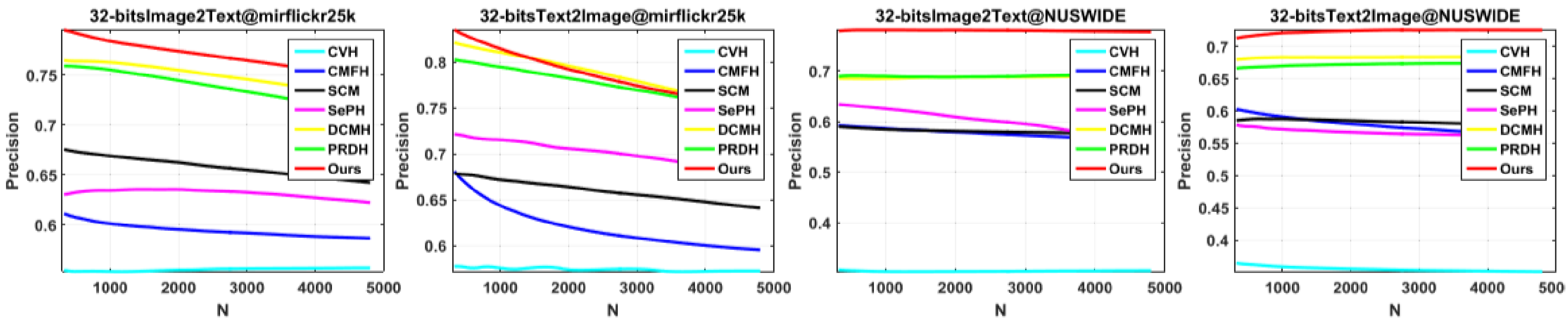


Fig. 3. Top$N$-precision curves. The baselines are based on hand-crafted features. The code length is 32.

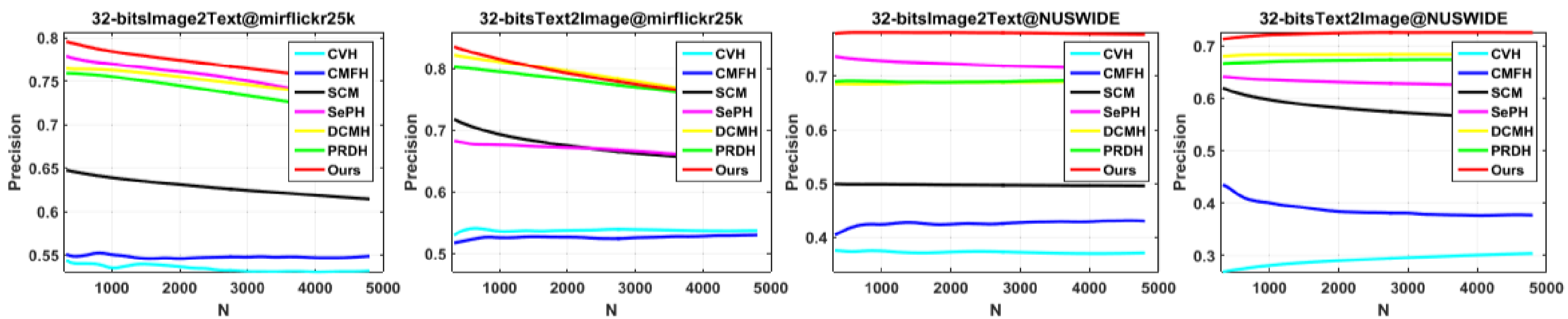


Fig. 5. Top$N$-precision curves. The baselines are based on CNN-F features. The code length is 32.

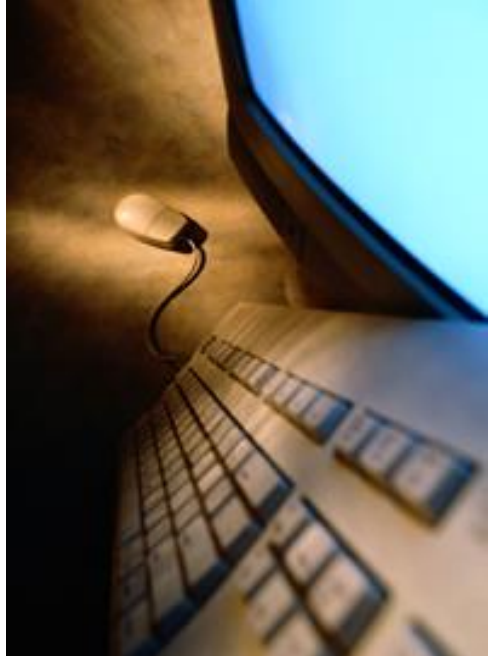
conclusion

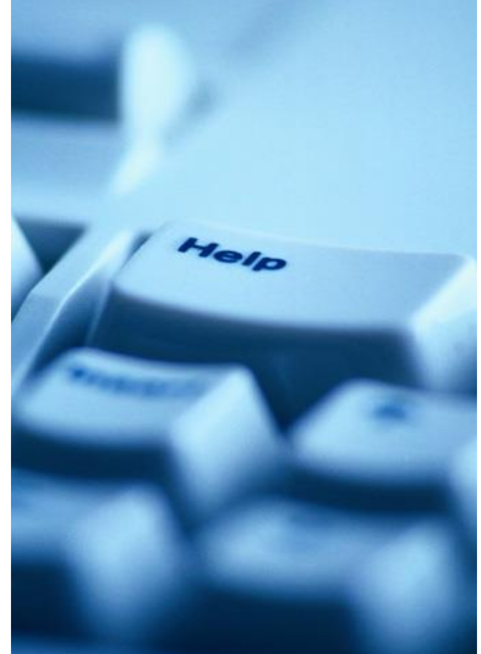
本文采用了triplet base作为训练输入，利用三元组的距离强化了不同类型之间距离的比较效果。
同时通过本文的实验结果，发现通过文本查询图像的精度更高，于是本文作者认为文本中存在更多的信息。

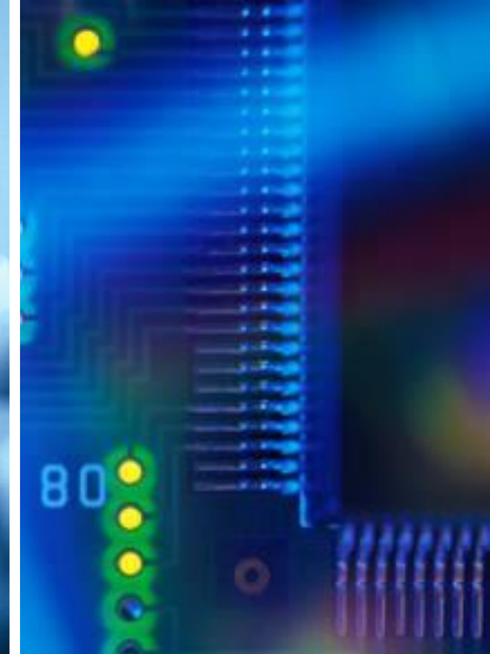
improve

本文对于图像的特征提取网络模型过于简单，导致认为图像中的语义包含较少。

由于CNN的特性，导致特征提取的空间关系较弱，建议采用更深的网络结构，同时融合不同感受野下的特征，增强特征空间关系，同时对于深层特征能够跟好的提取。另外应该抛弃全连接层，转而用1×1的卷积来代替，这样能够减少训练耗







谢谢！