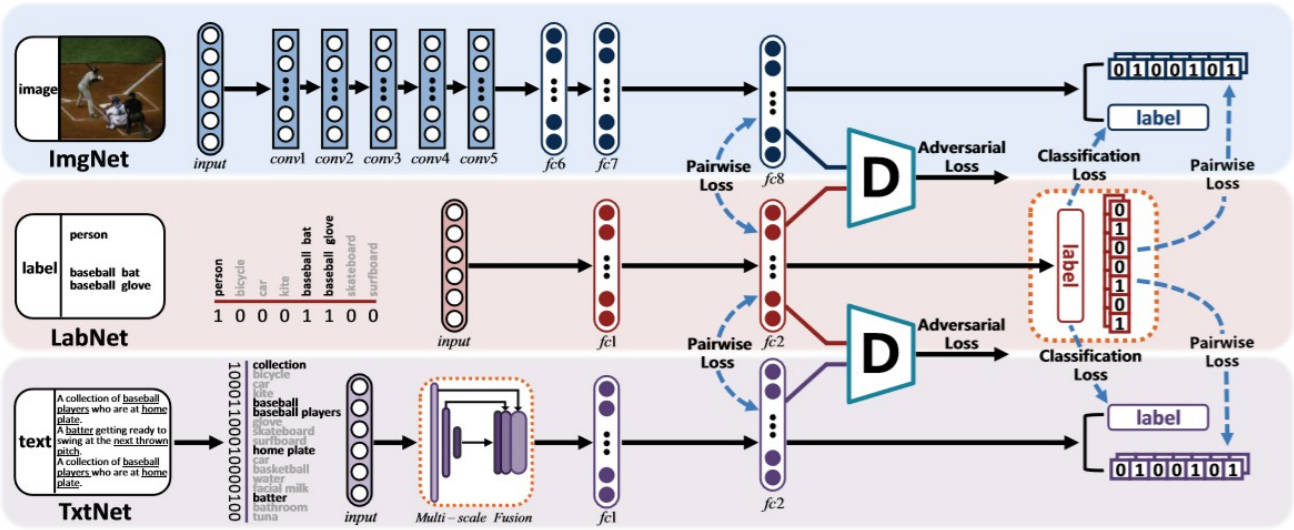
Self-Supervised Adversarial Hashing Networks for Cross-Modal Retrieval

from CVPR 2018

abstract

本文采用2个对抗网络来影响不同模态之间的相关性，并将标签作为训练样本，放入生成器生成对应的语义来供其他网络学习。本文采用multi-label的形式来表述label与对象之间的关系(即一个对象可能有多个label)。

network and loss



本文为了说明结果，采用了图像模态和文本模态的双模态分类。网络结构如上图，其中包含了两个特征提取网络：图像特征提取网络ImgNet与文本特征提取网络TxtNet。以及用来将label用于语义生成的labNet，还有结合ImgNet与labNet输出的辨别器D1，和结合TxtNet输出与LabNet输出的辨别器D2。

network struct

网络结构采用了目前CMH常见的网络模型。 ImgNet采用CNN-F模型，将最后的全连接层改为hash输出层，结构如下

Layer	Configuration
Conv1	64×11×11 s=4 pad=0 LRN，×2maxpool
Conv2	256×5×5 s=1 pad=2 LRN，×2maxpool
Conv3	256×3×3 s=1 pad=1
Conv4	256×3×3 s=1 pad=1
Conv5	256×3×3 s=1 pad=1 ,×2maxpool
Fc6	4096 dropout
Fc7	4096 dropout
Fc8	512 dropout
output	N

这里采用了其他论文中的描述，但与实际有些许不服，即最后fch8应为N，这里的N为hash code长度+分类标签个数，后续相同。 TxtNet使用了融合标签(MS)，每一层MS由一个average pooling + 1×1的conv构成，结构如下

Layer	Configuration
MS1	Pooling kernel=1
MS2	Pooling kernel=2
MS3	Pooling kernel=3
MS4	Pooling kernel=5
MS5	Pooling kernel=10
Fc1	4096
Fc2	512
Output	N

特征提取网络(ImgNet and TxtNet)均采用relu作为激活函数 LabNet采用全连接层的方式，即（label -> 4096 -> 512 -> N），采用sigmoid作为激活函数 对抗网络D(D1,D2相同)也是采用全连接层的方式，即(feature -> 4096 -> 4096 -> 1),采用tanh作为激活函数

loss

首先我们定义两个实例的相似度$S_{i,j}$，由于是多标签情况，所以只要两个实例有一个共同的标签，即表示两个实例是相似的，$S_{i,\ j}$=1，否之$S_{i,\ j}$=0. 其次，labNet的输入标签我们称为inputLabel，输出的分类标签label，其中label的维度在1×N ，与inputLabel的表示含义及其维度不同

我们用F^l表示$LabNet$的特征，F^v表示$ImgNet$输出的特征，F^t表示$TxtNet$输出的特征

我们用H^l表示$LabNet$的$hash$，H^v表示$ImgNet$输出的$hash$，H^t表示$TxtNet$输出的$hash$

对于$hash$的学习，我们定义$B^{v,t} \in \{-1,1\}^K$,这里K为$hash$的长度

semantic generation loss

$$
\begin{aligned}
\min_{B^l,\theta^l,\hat{L}^l} \mathcal{L}^l &= \alpha\mathcal{J}_1 + \gamma\mathcal{J}_2 + \eta\mathcal{J}_3 + \beta\mathcal{J}_4 \\
&= -\alpha \sum_{i,j=1}^{n} \left(S_{ij}\Delta_{ij}^l - \log\left(1 + e^{\Delta_{ij}^l}\right)\right) \\
&\quad -\gamma \sum_{i,j=1}^{n} \left(S_{ij}\Gamma_{ij}^l - \log\left(1 + e^{\Gamma_{ij}^l}\right)\right) \\
&\quad +\eta\left\|H^l - B^l\right\|_F^2 + \beta\left\|\hat{L}^l - L\right\|_F^2 \\
&\quad s.t. \quad B^l \in \{-1,1\}^K
\end{aligned}
\tag{3}
$$

其中$\Delta_{ij}^{l}=\frac{1}{2}(F_{*i}^{l})^{T}(F_{*j}^{l}),\Gamma_{ij}^{l}=\frac{1}{2}(H_{*i}^{l})^{T}(H_{*j}^{l}),\hat{L}^{l}$表示预测的分类$label$

这里的F^{l}为$labelNet$的$fc2$层的输出结果，即$F^{l}\in R^{1\times 512}$

H^{l}为$labelNet$的输出结果的前K部分，即输出的$hash\ code$

这里将该部分的loss分为4个部分，分别为J1相似实例间的语义loss，J2相似实例间的hashLoss，J3hash的学习loss，J4LabNet的分类loss。其中J1,J2为cross-modal的NLLloss定义。

feature loss

由于不同的模态采用不同的网络训练，所以我们用labNet生成的特征与两个网络中的特征进行相关联，这里并没有将输出进行融合，而是将接近输出层的全连接层的输出结果进行融合，这样既保证了特征的提取的有效性，又方便BP针对性的更新网络。

$$\begin{aligned}\min_{B^{v,t},\theta^{v,t}}\mathcal{L}^{v,t}=&\alpha\mathcal{J}_1+\gamma\mathcal{J}_2+\eta\mathcal{J}_3+\beta\mathcal{J}_4\\=&-\alpha\sum_{i,j=1}^{n}\left(S_{ij}\Delta_{ij}^{v,t}-\log\left(1+e^{\Delta_{ij}^{v,t}}\right)\right)\\&-\gamma\sum_{i,j=1}^{n}\left(S_{ij}\Gamma_{ij}^{v,t}-\log\left(1+e^{\Gamma_{ij}^{v,t}}\right)\right)\\&+\eta\|H^{v,t}-B^{v,t}\|_F^2+\beta\left\|\hat{L}^{v,t}-L\right\|_F^2\\&s.t.\quad B^{v,t}\in\{-1,1\}^K\end{aligned}\tag{4}$$

其中$\Delta_{ij}^{l}=\frac{1}{2}(F_{*i}^{l})^{T}(F_{*j}^{v(t)}),\Gamma_{ij}^{l}=\frac{1}{2}(H_{*i}^{l})^{T}(H_{*j}^{v(t)}),\hat{L}^{v(t)}$表示特征网络预测的分类$label$

这里的公式表示和上述相同，这里通过乘积将labNet的输出与特征提取网络的特征进行相关，利用labNet的自监督学习的结果作为可信的标签

adversarial loss

由于不同模态之间的间隔，我们通过labNet进行缩短间隔，那么用于缩短间隔的部分就是分辨网络D 我们定义两个实例的hash的Hamming距离为dis(bi,bj)=1/2(K-<bi,bj>),其中<bi,bj>为两个hash向量的内积运算，如果两个hash之间相似，那么内积的结果应该接近K(hash的长度,b中只包含-1和1两个值)

$$\min_{\theta_{adv}^{\star,l}}\mathcal{L}_{adv}^{\star,l}=\sum_{i=1}^{2\times n}\left\|D^{\star,l}(x_i^{\star l})-{y_i}^{\star,l}\right\|_2^2,\ \star=v,t\tag{5}$$

其中$x_i^{\star,l}$为图像(文本)语义与$labNet$语义的共同语义特征

$$y_i^{\star,l}=\begin{cases}0 & if\ labels\ for\ image(text)\\1 & if\ labels\ for\ inputLabel\end{cases}$$

这里对于每个D，使得D的输出接近于 真实的label的来源

optimization

本文中包含了3个hash code，但是为了训练，我们将三个hash code结合起来，称为B，定义如下

$$B = sign(H^v + H^t + H^l)$$

于是类似于GAN的loss，我们定义我们的总体损失函数为 生成器的loss-辨别器的loss即

$$\begin{aligned} \mathcal{L}_{gen} &= \mathcal{L}^v + \mathcal{L}^t + \mathcal{L}^l \\ \mathcal{L}_{adv} &= \mathcal{L}^v_{adv} + \mathcal{L}^t_{adv} \end{aligned} \tag{6}$$

If we put them together, we can obtain:

$$\begin{aligned} (B, \theta^{v,t,l}) &= \underset{B,\theta^{v,t,l}}{argmin}\ \mathcal{L}_{gen}(B, \theta^{v,t,l}) - \mathcal{L}_{adv}(\hat{\theta}_{adv}) \\ \theta_{adv} &= \underset{\theta_{adv}}{argmax}\ \mathcal{L}_{gen}(\hat{B}, \hat{\theta}^{v,t,l}) - \mathcal{L}_{adv}(\theta_{adv}) \\ s.t. &\quad B \in \{-1, 1\}^K \end{aligned} \tag{7}$$

其中$\hat{B}$表示B的单位向量

由于参数B的非连续性，我们采用分部的方式进行参数优化

Algorithm 1 Pseudopod showing the optimization of our SSAH

Require: Image set V ; Text set T ; Label set L ;
Ensure: Optimal code matrix B
Initialization
Initialize parameters: $\theta^{v,t,l}$, $\theta^{v,t}_{adv}$, α, γ, η, β
learnrate: μ, mini-batch size: $N^{v,t,l} = 128$, maximum iteration number: T_{max}.
repeat
  for t iteration **do**
    Update θ^l by BP algorithm:
    $\theta^l \leftarrow \theta^l - \mu \cdot \nabla_{\theta^l} \frac{1}{n} (\mathcal{L}_{gen} - \mathcal{L}_{adv})$
    Update the parameter $\theta^{v,t}$ by BP algorithm:
    $\theta^\star \leftarrow \theta^\star - \mu \cdot \nabla_{\theta^\star} \frac{1}{n} (\mathcal{L}_{gen} - \mathcal{L}_{adv}), \star = v, t$
    Update $\theta^\star_{adv}$ by BP algorithm:
    $\theta^\star_{adv} \leftarrow \theta^\star_{adv} - \mu \cdot \nabla_{\theta^\star_{adv}} \frac{1}{n} (\mathcal{L}_{(gen)} - \mathcal{L}_{adv}), \star = v, t$
  end for
  Update the parameter B by
  $B = sign(H + F + G)$
until convergence

1. 根据labNet的输出与分类的结果求取labNet的loss，对labNet的参数进行BP优化

2. 此时我们固定labNet的参数，通过labNet的输出来同样采用BP优化ImgNet和TxtNet
3. 最后我们固定labNet、imgNet、TxtNet的参数，利用他们的输出来更新俩个分辨器的参数
4. 重复1-3步，直到训练完整个数据集
5. 最后，当我们将所有的数据集训练后，我们根据训练后ImgNet、labNet、TxtNet的最终输出(hash code)来更新我们的B（binary code），然后进入下一次epoch重复上述1-4步

对于优化器的选择，本文选择了SGD(随机梯度下降)

training

hyperparameter

在训练中，作者对于生成器loss中的4个超参数进行测试，得出了最好的结果为

$$\alpha = \gamma = 1 \quad \eta = \beta = 10^{-4} \quad lr = 10^{-4} to 10^{-8}$$

performance

Table 2: MAP. The best accuracy is shown in boldface. The baselines are based on CNN-F features.

TASK	Method	Flickr-25K			NUS-WIDE			MS COCO		
		16 bits	32 bits	64 bits	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
I→ T	CVH [14]	0.557	0.554	0.554	0.400	0.392	0.386	0.412	0.401	0.400
	STMH [38]	0.602	0.608	0.605	0.522	0.529	0.537	0.422	0.459	0.475
	CMSSH [2]	0.585	0.584	0.572	0.511	0.506	0.493	0.512	0.495	0.482
	SCM [44]	0.671	0.682	0.685	0.533	0.548	0.557	0.483	0.528	0.550
	SePH [16]	0.657	0.660	0.661	0.478	0.487	0.489	0.463	0.487	0.501
	DCMH [12]	0.735	0.737	0.750	0.478	0.486	0.488	0.511	0.513	0.527
	OURS	**0.782**	**0.790**	**0.800**	**0.642**	**0.636**	**0.639**	**0.550**	**0.558**	**0.557**
T→ I	CVH [14]	0.557	0.554	0.554	0.372	0.366	0.363	0.367	0.359	0.357
	STMH [38]	0.600	0.606	0.608	0.496	0.529	0.532	0.431	0.461	0.476
	CMSSH [2]	0.567	0.569	0.561	0.449	0.389	0.380	0.429	0.408	0.398
	SCM [44]	0.697	0.707	0.713	0.463	0.462	0.471	0.465	0.521	0.548
	SePH [16]	0.648	0.652	0.654	0.449	0.454	0.458	0.449	0.474	0.499
	DCMH [12]	0.763	0.764	0.775	0.638	0.651	0.657	0.501	0.503	0.505
	OURS	**0.791**	**0.795**	**0.803**	**0.669**	**0.662**	**0.666**	**0.537**	**0.538**	**0.529**

Table 3: MAP. The best accuracy is shown in boldface. The baselines are based on vgg19 features.

TASK	Method	Flickr-25K			NUS-WIDE			MS COCO		
		16 bits	32 bits	64 bits	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
I→ T	CVH [14]	0.557	0.554	0.554	0.405	0.397	0.391	0.441	0.428	0.402
	STMH [38]	0.591	0.606	0.613	0.471	0.516	0.549	0.445	0.482	0.502
	CMSSH [2]	0.593	0.592	0.585	0.508	0.506	0.495	0.504	0.495	0.492
	SCM [44]	0.685	0.693	0.697	0.497	0.502	0.499	0.498	0.556	0.565
	SePH [16]	0.709	0.711	0.716	0.479	0.501	0.492	0.489	0.502	0.499
	DCMH [12]	0.677	0.703	0.725	0.590	0.603	0.609	0.497	0.506	0.511
	OURS	**0.797**	**0.809**	**0.810**	**0.636**	**0.636**	**0.637**	**0.550**	**0.577**	**0.576**
T→ I	CVH [14]	0.557	0.554	0.554	0.385	0.379	0.373	0.413	0.402	0.388
	STMH [38]	0.600	0.613	0.616	0.472	0.526	0.550	0.446	0.478	0.506
	CMSSH [2]	0.585	0.570	0.569	0.377	0.389	0.388	0.417	0.420	0.416
	SCM [44]	0.707	0.714	0.719	0.567	0.583	0.597	0.492	0.556	0.568
	SePH [16]	0.722	0.723	0.727	0.487	0.493	0.488	0.485	0.495	0.485
	DCMH [12]	0.705	0.717	0.724	0.620	0.634	0.643	0.507	0.520	0.527
	OURS	**0.782**	**0.797**	**0.799**	**0.653**	**0.676**	**0.683**	**0.552**	**0.578**	**0.578**

可以看出本文的SSAH结构比其他方法有明显改善，并且通过对于ImgNet的不同结构发现Vgg19的特征提取效果比CNN-F效果更好。

conclusion

本文将label作为输入之一，并且利用label的自监督学习生成的语义特征与其他模态的语义提取进行融合，利用对抗学习使得各模态与label的距离减小，利用label作为标准，拉近不同模态之间的距离。

缺点在于这里采用的mulit-label，该情况下的相似矩阵的设计过于简单，并且不能明显表明在有共同标签的对象间，标签数多的对象之间的相似性对比。

improvement

结合最近的DCMH方向的论文，本文可以对生成的hash code再次进行聚类，增加不同分类下hash code间的hamming距离 同时相似函数定义可以结合相关论文进行改进。