Triplet-Based Deep Hashing Network for Cross-Modal Retrieval

from IEEE 2018

abstract

本文通过对图像和文本采用不同的方式提取hash，通过hash对多模态对象进行分类的一种方法。 本文提出了通过query对象来提高损失函数的回归效率，类似于face net中的triplet loss；同时对于hash之间的hamming distance利用谱图回归的方式进行进一步聚类。

network and loss

1.network architecture

对于图像的hash提取网络，本文采用了CNN-F，仅对最后一层修改为hash code的提取层，网络结构如下

TABLE I

THE CNN ARCHITECTURE FOR IMAGE MODALITY

Layer	Configuration
conv1	kernel:64×11×11, stride:4, pad:0, LRN, ×2 pool
conv2	kernel:256×5×5, stride:1, pad:2, LRN, ×2 pool
conv3	kernel:256×3×3, stride:1, pad:1
conv4	kernel:256×3×3, stride:1, pad:1
conv5	kernel:256×3×3, stride:1, pad:1, ×2 pool
fc6	4096 dropout
fc7	4096 dropout
fch8	hash code length k

上图中 “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

对于文本的特征提取，则采用了非常简单的多层感知机模型，同样输出层由hash code提取层替换

TABLE II

THE MLP ARCHITECTURE FOR TEXTUAL MODALITY

Layer	Configuration
fc1	length of BOW vector
fc2	4096 dropout
fch3	hash code length k

2.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：

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

其中F表示图像

with

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

提取出的hash code，G表示文本提取出的hash code.

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.

这里α为需要调试的hyper-parameter.

本文中将loss分成3个部分，分别为inter-modal triplet loss, intra-modal triplet loss, graph regularization loss

inter-modal triplet loss

用来区分同一类的不同模态之间的hash code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 表示为：

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

Text to Image 的

inter-modal triplet loss 表示为：

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

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 = J_1 + J_2

intra-modal triplet loss

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采用如下计算方式：

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

对于文本类型对象

$$\text{where } \theta_{q_m^y p_m^y} = \frac{1}{2}\mathbf{F}_{*q_m}^{\top}\mathbf{F}_{*p_m} \text{ and } \theta_{q_m^y n_m^y} = \frac{1}{2}\mathbf{F}_{*q_m}^{\top}\mathbf{F}_{*n_m}.$$

采用如下计算方式：

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

最后得到intra-

$$\text{where } \theta_{q_m^x p_m^x} = \frac{1}{2}\mathbf{G}_{*q_m}^{\top}\mathbf{G}_{*p_m} \text{ and } \theta_{q_m^x n_m^x} = \frac{1}{2}\mathbf{G}_{*q_m}^{\top}\mathbf{G}_{*n_m}. \text{ Thus,}$$

modal triplet loss = J_3 + J_4

graph regularization loss

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

这里B^x B^y 分别是从文本模态和图像模态得出的hash code，$||A||_F$ 表示矩阵A的Frobenius norm，计算如下：

$$||A||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{i,j}|^2} = \sqrt{tr(A^H A)}$$

其中A^H为A的共轭矩阵

L为B的拉普拉斯矩阵（L = D - S 度矩阵 - 邻接矩阵）,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，可由下式表出：

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

where $\mathbf{S}$ is the similarity matrix and $\mathbf{B} = \{\mathbf{b}_i\}_{i=1}^{N}$ represents the unified hash codes. If $\mathbf{o}_i$ and $\mathbf{o}_j$ have the same labels, $s_{ij} = 1$; otherwise, $s_{ij} = 0$. We define $\mathbf{D} = diag(d_1, \ldots, d_n)$,

上述三种loss可以理解为：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 $$\min_{\mathbf{B},w_x,w_y} J = \min_{\mathbf{B},w_x,w_y} J_{inter} + J_{intra} + J_{re}. \quad (10)$$ 其中B是聚类的hash code， w_x 为text模态的权重，w_y 为image模态的权重

3.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。训练方式如下：

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;

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

updating B

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.

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

updating w_x

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

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

updating w_y

这里和上述类似，公式如下：

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

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

performance

本文对于该方法的表现，发现文本任务的结果比图像任务的结果更好，认为图像中包含的语义较文本更少。

improve

个人认为，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，同时对于MPL不能很好的检测出文本的语义时间序列关系。

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时。

对于MPL，本文中的模型虽然较CNN-F能够提取出较多的特征，但由于文本的特性，其特征简单易提取，但MPL有可能忽略的文本中上下文关系，建议采用LSTM结构，对文本特征进行上下文的特征结合，获取更精确的特征。 关于Text的特征提取，建议参考《Cross-Modal Scene Networks》 IEEE 2018