Cycle-Consistent Deep Generative Hashing for Cross-Modal Retrieval

from TIP 2019

abstract

由于生成的hash的对于输入的对象会有信息丢失，本文利用cycle-GAN，通过将生成的hash返回生成输入，从而减少在有模态对象生成hash code的过程中信息的丢失。同时利用随机hash生成结合概率模型获得loss函数。

introduction

由于很多实际因素，成对的训练数据在未来将难以实现，所以本文接用DGH(Deep Generative Hashing)将不同模态对象的提取的特征与hashing进行相互生成，从而减少因feature map到hashing之间的信息损失。同时结合cycle-consistency，将不同模态之间的对象特征进行相互转换，利用GAN的原理，增强分类器对于来自不同模态特征所生成的该模态下特征的分类能力。以图像与文本两个模态为例。

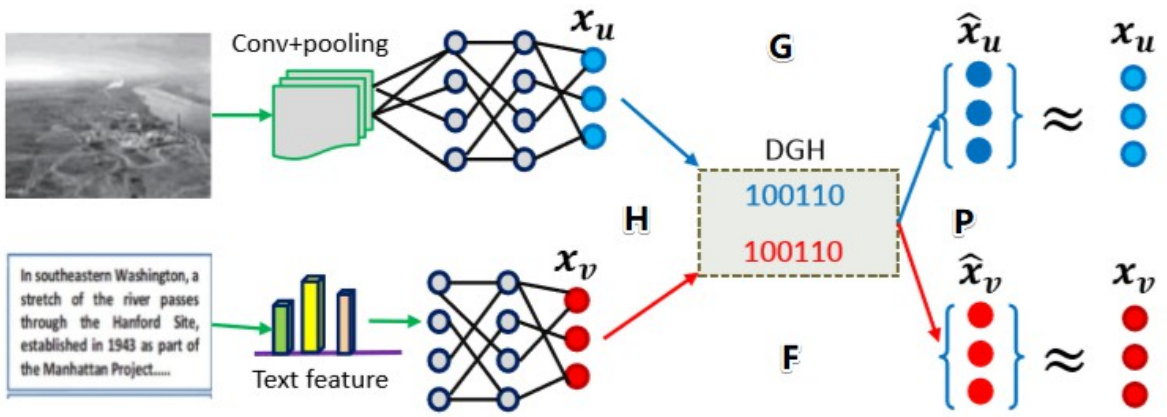


Fig. 1: The architecture overview of our proposed CYC-DGH for cross-modal retrieval. Given two modalities of images and texts, we learn a couple of generative hash functions in the absence of paired correspondence through the cycle-consistent adversarial learning. This is achieved by training two mappings: $\boldsymbol{G} : \boldsymbol{X}_u \rightarrow \boldsymbol{X}_v$ and $\boldsymbol{F} : \boldsymbol{X}_v \rightarrow \boldsymbol{X}_u$, which are reverse of each other, and adding a cycle consistency loss into the adversarial loss yields $\boldsymbol{F}(\boldsymbol{G}(\boldsymbol{x}_u)) \approx \boldsymbol{x}_u$, and $\boldsymbol{G}(\boldsymbol{F}(\boldsymbol{x}_v)) \approx \boldsymbol{x}_v$. The deep generative model performs both hash function learning and regeneration inputs from binary codes. See text for details.

按照上述图中的描述过程，映射G表示由图像特征生成文本特征，F表示由文本特征生成图像特征。其中映射的过程包含由输入特征到hashing的映射H，以及由H映射出的hashing对输入特征的反映射P，最后再由P反映射出的输入特征来映射生成另外一个模态的特征。具体过程在上图中标出。

Network and training objective

network structure

本文涉及到3个过程：

1. 不同模态对象的特征提取网络
2. feature map与hash互相生成网络
3. 不同模态feature map的互相生成与辨别网络

在文中明确说明了文本模态的特征提取网络为“Generative Adversarial Text to Image Synthesis”中的文本编码模型，从该参看文献中引用对于该模型的描述。

> The description embedding $\phi(t)$ is first compressed using a fully-connected layer to a small dimension (in practice we used 128) followed by leaky-ReLU and then concatenated to the noise vector z

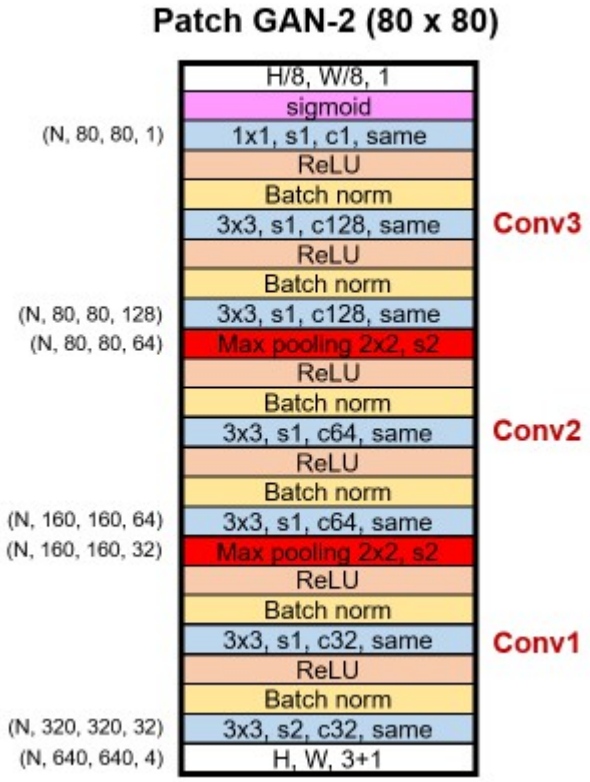
同时本文中对于不同训练集进行了不同的模型修改，结构如下

Dataset	Layer	config
coco	FC1	1000
	FC2	500
	FC3	200
IAPR TC-12	FC1	11500
	FC2	500
	FC3	200
Wiki	FC1	10
	FC2	500
	FC3	200
	FC1	128 with leaky relu

图像生成的网络使用了在风格迁移上有很好效果的“Perceptual Losses for Real-Time Style Transfer and Super-Resolution”中的结构生成网络结构，结构如下

Layer	config
Conv1	Kernel = 9, stride=1, out=32
Conv2	Kernel = 3, stride=2, out=64
Conv3	Kernel = 3, stride=2, out=128
Residual Block several	Kernel = 3, stride=1, out=128, num=5
Deconv1	Kernel = 3, stride=1/2, out=64
Deconv2	Kernel = 3, stride=1/2, out=32
Deconv3	Kernel = 3, stride=1, out=3

辨别网络选择“Image-to-Image Translation with Conditional Adversarial Networks”中的70×70的PatchGAN网络结构



此外，文章在实验中采用了CNN-F，fc7的输出作为图像的提取特征。

loss and training objective

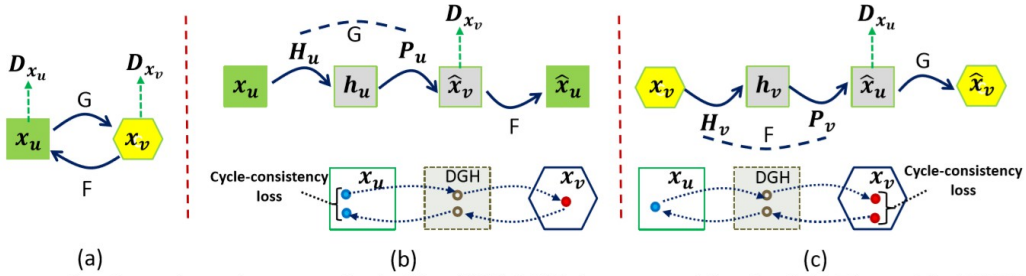


Fig. 3: The proposed cycle-consistent deep generative hashing (CYC-DGH) for cross-modal retrieval. (a) The model of CYC-DGH couples two mappings: $\boldsymbol{G}: \boldsymbol{x}_u \rightarrow \boldsymbol{x}_v$ and $\boldsymbol{F}: \boldsymbol{x}_v \rightarrow \boldsymbol{x}_u$ as well as associated adversarial discriminators $\boldsymbol{D}_{x_v}$ and $\boldsymbol{D}_{x_u}$. The two mappings are decomposed into the binary code generation and the reverse process of regenerating inputs from binary codes: $\boldsymbol{G}: \boldsymbol{x}_u \rightarrow \boldsymbol{H}_u \rightarrow \boldsymbol{P}_u \rightarrow \boldsymbol{x}_v$ and $\boldsymbol{F}: \boldsymbol{x}_v \rightarrow \boldsymbol{H}_v \rightarrow \boldsymbol{P}_v \rightarrow \boldsymbol{x}_u$. To regulate the mappings, two cycle-consistent losses are introduced: (b) forward $\boldsymbol{x}_u \rightarrow G(\boldsymbol{x}_u) \rightarrow \boldsymbol{F}(\boldsymbol{G}(\boldsymbol{x}_u)) \approx \hat{\boldsymbol{x}}_u$, and (c) backward $\boldsymbol{x}_v \rightarrow F(\boldsymbol{x}_v) \rightarrow \boldsymbol{G}(\boldsymbol{F}(\boldsymbol{x}_v)) \approx \hat{\boldsymbol{x}}_v$.

adversarial loss

adversarial loss为图像特征与文本特征互相生成的loss。G表示图像特征生成文本特征，F表示文本特征生成图像特征，x_U表示图像特征，x_V表示文本特征。Dx_U表示用于区分生成的图像特征{$F(x_V)$}与原生图片特征{x_U}，Dx_V表示用于区分生成的文本特征{$F(x_U)$}与原生文本特征{x_V}.其中G和H的定义在introduction中有介绍，其中的H,P两个过程的loss将在后续定义。

$$L_{GAN}(G, D_{x_v}, x_u, x_v) = E_{x_v \sim p_{data}(x_v)}[logD_{x_v}(x_v)] + E_{x_u \sim p_{data}(x_u)}[log(1 - D_{x_v}(G(x_u)))]$$

$$L_{GAN}(G, D_{x_u}, x_v, x_u) = E_{x_u \sim p_{data}(x_u)}[logD_{x_u}(x_u)] + E_{x_v \sim p_{data}(x_v)}[log(1 - D_{x_u}(G(x_v)))]$$

这里与一般的GAN不同的是，loss并没有采用nll loss，而是采用了least-square loss（最小二乘法loss），

$$train\ for\ generator\ to\ minimize: E_{x_v \sim p_{data}(x_v)}[(D_{x_v}(x_v) - 1)^2]$$

$$train\ for\ distinguish\ to\ minimize: E_{x_u \sim p_{data}(x_u)}[(D_{x_v}(x_u)-1)^2] + : E_{x_v \sim p_{data}(x_v)}[D_x(x_v)]$$

cycle-consistency loss

cycle-consistency loss用来表示某一模态生成指定模态的特征后再由其生成会原本模态的特征，该生成的特征与原有的特征的距离，即由txt生成的img'，由img'生成txt'，对比txt与txt'之间的距离，定义如下：

$$L_{cyc}(G,F) = E_{x_v \sim p_{data}(x_u)}[||F(G(x_u)) - x_u||_1] + E_{x_u \sim p_{data}(x_u)}[||G(F(x_v)) - x_u||_1]$$

deep generative hashing

这里考虑的是由不同模态间的feature map生成hash 与 hash 生成 feature map过程产生的loss

1. hash 生成 feature map

我们先做如下定义：

$$P_u : h_u \rightarrow x_v,\ denoted\ as\ p(x_v|h_u) \qquad P_v : h_v \rightarrow x_u,\ denoted\ as\ p(x_u|h_v)$$

由SGH("Stochastic Generative Hashing")中提出的内容，我们引用如下内容

We use a simple Gaussian distribution to model the generation of x given h:

$$p(x,h) = p(x|h)p(h),\ \text{where } p(x|h) = \mathcal{N}(Uh, \rho^2 I) \tag{1}$$

and $U = \{u_i\}_{i=1}^l$, $u_i \in \mathbb{R}^d$ is a codebook with l codewords. The prior $p(h) \sim \mathcal{B}(\theta) = \prod_{i=1}^l \theta_i^{h_i}(1-\theta_i)^{1-h_i}$ is modeled as the multivariate Bernoulli distribution on the hash codes, where $\theta = [\theta_i]_{i=1}^l \in [0,1]^l$. Intuitively, this is an additive model which reconstructs x by summing the selected columns of U given h, with a Bernoulli prior on the distribution of hash codes. The joint distribution can be written as:

$$p(x,h) \propto \exp\left(\frac{1}{2\rho^2} \underbrace{(x^\top x + h^\top U^\top U h - 2x^\top U h)}_{\|x - U^\top h\|_2^2} - (\log \frac{\theta}{1-\theta})^\top h \right) \tag{2}$$

其中高斯重构误差为$||x - U^T h||_2^2$表示欧式领域稳定程度，当范数U是有限的时候，误差越小表示稳定性越高

所以我们只需要保证高斯重构误差 -logp(x|h)最小化，即可保证生成的feature map与原有的feature map之间领域稳定

2. feature map 生成 hash

由目前再自动编码上的研究，在概率模型p(h|x)上寻找最优解是很难的，所以这里依旧借助SGH里的内容进行定义

Even with the simple Gaussian model (1), computing the posterior $p(h|x) = \frac{p(x,h)}{p(x)}$ is not tractable, and finding the MAP solution of the posterior involves solving an expensive integer programming subproblem. Inspired by the recent work on variational auto-encoder (Kingma and Welling, 2013; Mnih and Gregor, 2014; Gregor et al., 2014), we propose to bypass these difficulties by parameterizing the encoding function as

$$q(h|x) = \prod_{k=1}^l q(h_k = 1|x)^{h_k} q(h_k = 0|x)^{1-h_k}, \tag{3}$$

to approximate the exact posterior $p(h|x)$. With the linear parametrization, $h = [h_k]_{k=1}^l \sim \mathcal{B}(\sigma(W^\top x))$ with $W = [w_k]_{k=1}^l$. At the training step, a hash code is obtained by sampling from $\mathcal{B}(\sigma(W^\top x))$. At the inference step, it is still possible to sample h. More directly, the MAP solution of the encoding function (3) is readily given by

$$h(x) = \underset{h}{\text{argmax}}\, q(h|x) = \frac{\text{sign}(W^\top x) + 1}{2}$$

This involves only a linear projection followed by a sign operation, which is common in the hashing literature. Computing $h(x)$ in our model thus has the same amount of computation as ITQ (Gong and Lazebnik, 2011), except without the orthogonality constraints.

$$q(h|x) = \prod_{k=1}^{K} q(h_k = 1|x)^{h_k} q(h_k = 0|x)^{1-h_k}$$

$$其中 h = [h_k]_{k=1}^{K} \sim B(\sigma(W^T x)) 是线性参数化的，其中 W = [w_k]_{k=1}^{K}$$

然后结合W进行优化，后得到优化后的结果

$$p(h|x) = arg\max_{h} q(h|x) = \frac{sign(W^T x) + 1}{2}$$

training objective

最终训练的目标函数即cycle-gan部分得loss + hash 重构部分得loss，即

$$L(G, F, D_{x_u}, D_{x_v}, H_*) = L_{GAN}(G, D_{X_v}, x_u, x_v) + L_{GAN}(G, D_{x_u}, x_v, x_u) + \lambda L_{cyc}(G, F) + D_{KL}(q(h_*|x_*)||p(h_*|x_*)) + L(\theta_*; x_*)$$

$$其中 * = \{u, v\}, D_{KL}(p||q) = \sum_{x \in X}[p(x)logp(x) - p(x)logq(x)],$$

$$L(\theta_*; x_*) = E_{q(h*|x_x)}[-logq(h_*|x_*) + logp(x_*|h_*)] 其中 \theta_* = \{W_*, U_*, \rho_*, \beta_* := log\frac{\theta}{1-\theta}\}$$

由于目标函数关于p(x|h)得导数难以获得，所以本文依旧利用了SGH中得内容，将上式中得h进行替换，内容如下

In next section, we first provide an *unbiased* estimator of the gradient w.r.t. W derived based on distributional derivative, and then, we derive a *simple* and *efficient* approximator. Before we derive the estimator, we first introduce the stochastic neuron for reparametrizing Bernoulli distribution. A stochastic neuron reparameterizes each Bernoulli variable $h_k(z)$ with $z \in (0, 1)$. Introducing random variables $\xi \sim \mathcal{U}(0, 1)$, the stochastic neuron is defined as

$$\tilde{h}(z, \xi) := \begin{cases} 1 & \text{if } z \geqslant \xi \\ 0 & \text{if } z < \xi \end{cases}. \tag{5}$$

Because $\mathbb{P}(\tilde{h}(z, \xi) = 1) = z$, we have $\tilde{h}(z, \xi) \sim \mathcal{B}(z)$. We use the stochastic neuron (5) to reparameterize our binary variables h by replacing $[h_k]_{k=1}^{l}(x) \sim \mathcal{B}(\sigma(w_k^\top x))$ with $[\tilde{h}_k(\sigma(w_k^\top x), \xi_k)]_{k=1}^{l}$. Note that $\tilde{h}$ now behaves deterministically given ξ. This gives us the reparameterized version of our original training objective (4):

$$\tilde{H}(\Theta) = \sum_x \tilde{H}(\Theta; x) := \sum_x \mathbb{E}_\xi \left[\ell(\tilde{h}, x)\right], \tag{6}$$

where $\ell(\tilde{h}, x) := -\log p(x, \tilde{h}(\sigma(W^\top x), \xi)) + \log q(\tilde{h}(\sigma(W^\top x), \xi)|x)$ with $\xi \sim \mathcal{U}(0, 1)$. With such a reformulation, the new objective can now be optimized by exploiting the distributional stochastic gradient descent, which will be explained in the next section.

即通过h^将之前得h进行了收缩，使得能够通过计算SGD来代替之前得不连续分布

training

本文在训练阶段中，所有经过辨别器D的数据都是之前生成的数据(可靠性高)，不是新生成的数据，这样能够保证网络的稳定性。同时超参数λ=10，学习率从前100epoch的0.0002动态降低至0

experiments

本文选用了COCO，IAPR TC-12, Wiki数据集，同时进行单独测试，就仅有cycle的精度和仅有Gan的精度进行对比，实验结果如下

Loss	Per-class accuracy	Per-pixel accuracy
Cycle alone	0.270	0.724
GAN alone	0.611	0.126
CYC-DGH	**0.584**	**0.192**

TABLE I: FCN-scores for different variants of CYC-DGH, evaluated on Microsoft-COCO with 64 bits to regenerate the images.

结果也证明两者对不同的评估方向效果不同，如果想保证精度，就需要两者进行权衡，刚刚提到的权衡比例λ=10

同时本文还证明了与ITQ(Iterative quantization)方法的速度比较，其结果表明CYC-DGH完爆ITQ的速度

Loss	Per-class accuracy	Per-pixel accuracy
Cycle alone	0.270	0.724
GAN alone	0.611	0.126
CYC-DGH	**0.584**	**0.192**

TABLE I: FCN-scores for different variants of CYC-DGH, evaluated on Microsoft-COCO with 64 bits to regenerate the images.

精度对比上也是较其他方法由明显提高

TABLE V: Mean Average Precision (MAP) comparison of state-of-the-art cross-modal hashing methods on three data sets.

Task	Method	Microsoft COCO				IAPR TC-12				Wiki			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
$I \rightarrow T$	CVH [35]	0.373	0.368	0.366	0.357	0.537	0.541	0.524	0.496	0.238	0.204	0.179	0.158
	SCM [19]	0.570	0.600	0.631	0.649	0.567	0.505	0.454	0.418	0.139	0.137	0.141	0.136
	LSSH [36]	-	-	-	-	0.544	0.577	0.596	0.599	0.364	0.371	0.378	0.358
	SePH [1]	0.581	0.613	0.625	0.634	0.618	0.645	0.650	0.678	0.414	0.435	0.437	0.447
	CYC-DGH	**0.722**	**0.754**	**0.781**	**0.780**	**0.771**	**0.815**	**0.832**	**0.831**	**0.794**	**0.811**	**0.813**	**0.820**
$T \rightarrow I$	CVH [35]	0.373	0.369	0.365	0.371	0.568	0.578	0.561	0.536	0.388	0.336	0.257	0.230
	SCM [19]	0.558	0.619	0.658	0.686	0.652	0.570	0.478	0.421	0.132	0.143	0.156	0.149
	LSSH [36]	-	-	-	-	0.487	0.526	0.555	0.572	0.606	0.626	0.638	0.638
	SePH [1]	0.613	0.650	0.672	0.693	0.610	0.634	0.640	0.673	0.701	0.699	0.710	0.715
	CYC-DGH	**0.761**	**0.796**	**0.834**	**0.859**	**0.772**	**0.798**	**0.837**	**0.842**	**0.811**	**0.823**	**0.826**	**0.822**

conclusion