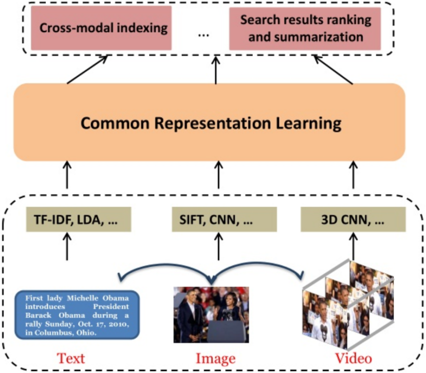
Deep Joint-Semantics Reconstructing Hashing for Large-Scale Unsupervised Cross-Modal Retrieval

Oriol Vinyals[1] Zhisheng Zhong[2]

[1]Key Laboratory of Machine Perception (MOE), School of EECS, Peking University

[2]Key Laboratory of Machine Perception (MOE), School of EECS, Peking University

Cross-Modal Retrieval



Binary representation learning
Unsupervised Cross-Modal Retrieval

Contributions

- Put forward affinity matrix.
- Reconstruct above jointsemantics, friendly for batch-wise training.
- Reach good result.

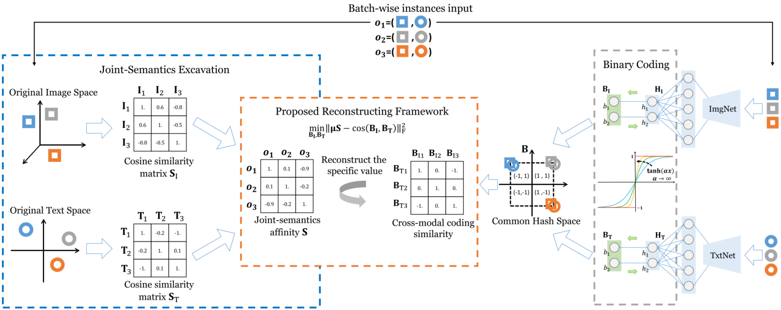


图: The pipeline of DJSRH

Defination

- m: Batch size;
- $\mathcal{O}$: $\{o_k = \lceil \mathsf{I}_k, \mathsf{T}_k \rfloor\}_{k=1}^{m}$, include each image-text pair. Feature matrix of image and text are defined as $\mathsf{F}_\mathsf{I} \in \mathbb{R}^{m \times p_I}$ and $\mathsf{F}_\mathsf{T} \in \mathbb{R}^{m \times p_T}$;
- $\mathsf{B}_\mathsf{I} \in \{\pm 1\}^{m \times d}$ and $\mathsf{B}_\mathsf{T} \in \{\pm 1\}^{m \times d}$: Binary repesentation given out by ImgNet and TxtNet from the input I_k and T_k;
- $\hat{\mathsf{F}}_\mathsf{I}$ and $\hat{\mathsf{F}}_\mathsf{T}$: The normalized F_I and F_T, the cosine similarity matrices

$$\mathsf{S}_\mathsf{I} = \hat{\mathsf{F}}_\mathsf{I} \hat{\mathsf{F}}_\mathsf{I}^\top \in [-1, +1]^{m \times m}$$

$$\mathsf{S}_\mathsf{T} = \hat{\mathsf{F}}_\mathsf{T} \hat{\mathsf{F}}_\mathsf{T}^\top \in [-1, +1]^{m \times m}$$

Constructing Joint-Semantics Matrix

Laplacian constrains

$$\min_{\mathsf{B}} \beta \mathrm{Tr}(\mathsf{B}^\top \mathsf{L}_\mathsf{I} \mathsf{B}) + (1-\beta)\mathrm{Tr}(\mathsf{B}^\top \mathsf{L}_\mathsf{T} \mathsf{B}) \quad \text{s.t. } \mathsf{B} \in \{\pm 1\}^{m \times d}$$

where

$$\mathsf{L}_\mathsf{I} = \mathrm{diag}(\mathsf{S}_1 1) - \mathsf{S}_\mathsf{I}$$

$$\mathsf{L}_\mathsf{T} = \mathrm{diag}(\mathsf{S}_\mathsf{T} 1) - \mathsf{S}_\mathsf{T}$$

are Laplacian matrices.

Constructing Joint-Semantics Matrix

Joint-semantics Affinity Matrix

Define $\mathcal{C}$ as combination function, then

$$\mathsf{S} = \mathcal{C}(\mathsf{S}_\mathsf{I}, \mathsf{S}_\mathsf{T}) \in [-1, +1]^{m \times m}$$

Merge Img and Txt as

$$\tilde{\mathsf{S}} = \beta \mathsf{S}_\mathsf{I} + (1-\beta)\mathsf{S}_\mathsf{T}$$

Then

$$\begin{aligned} \mathsf{S} &= \mathcal{C}(\mathsf{S}_\mathsf{I}, \mathsf{S}_\mathsf{T}) \\ &= (1-\eta)\tilde{\mathsf{S}} + \eta \frac{\tilde{\mathsf{S}}\tilde{\mathsf{S}}^\top}{m} \\ &= (1-\eta)[\beta \mathsf{S}_\mathsf{I} + (1-\beta)\mathsf{S}_\mathsf{T}] + \frac{\eta}{m}[\beta^2 \mathsf{S}_\mathsf{I}{\mathsf{S}_\mathsf{I}}^\top + \beta(1-\beta)(\mathsf{S}_\mathsf{I}{\mathsf{S}_\mathsf{T}}^\top + \mathsf{S}_\mathsf{T}{\mathsf{S}_\mathsf{I}}^\top) \end{aligned}$$

S_{ij} indicates the latent semantic similarity between o_i 和 o_j.

Object

$$\min_{\mathsf{B_I},\mathsf{B_T}} ||\mu \mathsf{S} - \cos(\mathsf{B_I}, \mathsf{B_T})||_{\mathsf{F}}^2, \quad \text{s.t. } \mathsf{S} = \mathcal{C}(\mathsf{S_I}, \mathsf{S_T}) \in [-1, +1]^{m \times m}$$

Laplacian constrains

$$\mathsf{Tr}(\mathsf{B}^\top \mathsf{L} \mathsf{B}) = \sum_{i,j} \mathsf{S}_{ij} ||\mathsf{B}_i - \mathsf{B}_j||^2$$

Object with intra-modal influence

$$\begin{aligned} \min_{\mathsf{B_I},\mathsf{B_T}} ||\mu \mathsf{S} - \cos(\mathsf{B_I}, \mathsf{B_T})||_{\mathsf{F}}^2 + \lambda_1 ||\mu \mathsf{S} - \cos(\mathsf{B_I}, \mathsf{B_I})||_{\mathsf{F}}^2 \\ + \lambda_2 ||\mu \mathsf{S} - \cos(\mathsf{B_T}, \mathsf{B_T})||_{\mathsf{F}}^2, \\ \text{s.t. } \mathsf{S} = \mathcal{C}(\mathsf{S_I}, \mathsf{S_T}) \in [-1, +1]^{m \times m}, \ \mathsf{B_I}, \mathsf{B_T} \in \{-1, +1\}^{m \times d} \end{aligned}$$

Optimization

Set $\mathsf{H} \in \mathbb{R}^{m \times d}$ as the last layer of ImgNet and TxtNet without activate function, then

$$\mathsf{B} = \mathrm{sgn}(\mathsf{H}) \in \{-1, +1\}^{m \times d}$$

Use the following instead,

$$\mathsf{B} = \tanh(\alpha \mathsf{H}) \in \{-1, +1\}^{m \times d}, \ \alpha \in \mathbb{R}^{+}$$

Algorithm

Algorithm 1 Deep Joint-Semantics Reconstructing Hashing

Input:

Training set $\{\boldsymbol{o}_k = [\mathbf{I}_k, \mathbf{T}_k]\}_{k=1}^{n}$ and their corresponding original features $\mathbf{F}_\mathrm{I}$ and $\mathbf{F}_\mathrm{T}$; ImgNet $\mathcal{G}_{\theta_\mathrm{I}}$ and TxtNet $\mathcal{G}_{\theta_\mathrm{T}}$ with θ_I and θ_T denoting the deep network parameters; batch size m;

Output:

Hashing coding function $\varphi_\mathrm{I}(x) = \mathrm{sgn}(\mathcal{G}_{\theta_\mathrm{I}}(x))$ for image input and $\varphi_\mathrm{T}(x) = \mathrm{sgn}(\mathcal{G}_{\theta_\mathrm{T}}(x))$ for text input;

```
Initialize epoch t = 0;
repeat
    t = t + 1;  α = √t;
    for ⌊n/m⌋ iterations do
        Randomly sample a batch of instances from training set {o_k = [I_k, T_k]}_{k=1}^m;
        Calculate the normalized F̂_I, F̂_T and integrate the cosine matrices S_I = F̂_I F̂_I^⊤, S_T = F̂_T F̂_T^⊤ to the joint-semantics affinity S with Equation (3);
        Forward propagate H_I = G_θI(I), H_T = G_θT(T);
        Hash coding with activation function (7) B_I = tanh(αH_I), B_T = tanh(αH_T);
        Calculate the objective function (5), back propagate the gradients with the chain rule and update the whole parameters;
    end for
until convergence
```

Algorithm

```
F_I = F.normalize(F_I)
S_I = F_I.mm(F_I.t())
S_I = S_I * 2 - 1

F_T = F.normalize(F_T)
S_T = F_T.mm(F_T.t())
S_T = S_T * 2 - 1


B_I = F.normalize(code_I)
B_T = F.normalize(code_T)

BI_BI = B_I.mm(B_I.t())
BT_BT = B_T.mm(B_T.t())
BI_BT = B_I.mm(B_T.t())

S_tilde = settings.BETA * S_I + (1 - settings.BETA) * S_T
S = (1 - settings.ETA) * S_tilde + settings.ETA * S_tilde.mm(S_tilde) / settings.BATCH_SIZE
S = S * settings.MU

loss1 = F.mse_loss(BI_BI, S)
loss2 = F.mse_loss(BI_BT, S)
loss3 = F.mse_loss(BT_BT, S)
loss = settings.LAMBDA1 * loss1 + 1 * loss2 + settings.LAMBDA2 * loss3
```

Experiment

Task	Method	Wiki				MIRFlickr				NUS-WIDE			
		16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits
$I \rightarrow T$	CVH	0.179	0.162	0.153	0.149	0.606	0.599	0.596	0.598	0.372	0.362	0.406	0.390
	IMH	0.201	0.203	0.204	0.195	0.612	0.601	0.592	0.579	0.470	0.473	0.476	0.459
	CMFH	0.251	0.253	0.259	0.263	0.621	0.624	0.625	0.627	0.455	0.459	0.465	0.467
	LSSH	0.197	0.208	0.199	0.195	0.584	0.599	0.602	0.614	0.481	0.489	0.507	0.507
	DBRC	0.253	0.265	0.269	0.288	0.617	0.619	0.620	0.621	0.424	0.459	0.447	0.447
	UDCMH	0.309	0.318	0.329	0.346	0.689	0.698	0.714	0.717	0.511	0.519	0.524	0.558
	DJSRH	**0.388**	**0.403**	**0.412**	**0.421**	**0.810**	**0.843**	**0.862**	**0.876**	**0.724**	**0.773**	**0.798**	**0.817**
$T \rightarrow I$	CVH	0.252	0.235	0.171	0.154	0.591	0.583	0.576	0.576	0.401	0.384	0.442	0.432
	IMH	0.467	0.478	0.453	0.456	0.603	0.595	0.589	0.580	0.478	0.483	0.472	0.462
	CMFH	0.595	0.601	0.616	0.622	0.642	0.662	0.676	0.685	0.529	0.577	0.614	0.645
	LSSH	0.569	0.593	0.593	0.595	0.637	0.659	0.659	0.672	0.577	0.617	0.642	0.663
	DBRC	0.574	0.588	0.598	0.599	0.618	0.626	0.626	0.628	0.455	0.459	0.468	0.473
	UDCMH	**0.622**	0.633	0.645	**0.658**	0.692	0.704	0.718	0.733	0.637	0.653	0.695	0.716
	DJSRH	0.611	**0.635**	**0.646**	**0.658**	**0.786**	**0.822**	**0.835**	**0.847**	**0.712**	**0.744**	**0.771**	**0.789**

图: mAP@50

Experiment

MIRFlickr, CODELEN $= 64$

- mAP of Image to Text: 0.865
- mAP of Text to Image: 0.853