Learning to Hash with Binary Deep Neural Network

Thanh-Toan Do, Anh-Dzung Doan, Ngai-Man Cheung

Singapore University of Technology and Design

{thanhtoan_do, dung_doan, ngaiman_cheung}@sutd.edu.sg

Hashing for Image Search

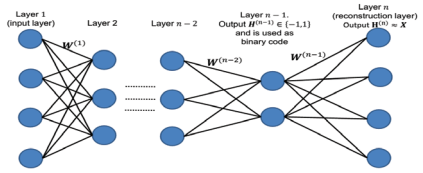
- Nearest neighbor search in database of N data points in $\mathbb{R}^D$
  - Exact search: $\mathcal{O}(ND)$ in both time and space → difficult for handling large scale, e.g., million SIFT features in 3D model.
- Finding nearest neighbors in Hamming space is more efficient
  - Time and space: $\mathcal{O}(NL)$ instead of $\mathcal{O}(ND)$
  - Hamming distance by hardware operation, i.e., XOR.

Contributions

- Define hash function as a deep network
- ***In order to deal with binary constraints, instead of involving the sgn or step function as in previous work, our design constrains one layer to directly output the binary codes***
- Propose a framework to solve the resulting NP-hard optimization with binary constraints
- Directly incorporate the independence and balance properties on the codes
- Ensure the similarity preserving

→ *Binary Deep Neural Network* for both unsupervised and supervised hashing

Unsupervised Hashing



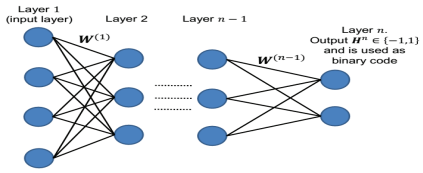
$$\min_{\mathbf{W},\mathbf{c},\mathbf{B}} \frac{1}{2m}\left\|\mathbf{X}-\mathbf{W}^{(n-1)}\mathbf{B}-\mathbf{c}^{(n-1)}\mathbf{1}_{1\times m}\right\|^2+\frac{\lambda_1}{2}\sum_{l=1}^{n-1}\left\|\mathbf{W}^{(l)}\right\|^2$$

$$+\frac{\lambda_2}{2m}\left\|\mathbf{H}^{(n-1)}-\mathbf{B}\right\|^2+\frac{\lambda_3}{2}\left\|\frac{1}{m}\mathbf{H}^{(n-1)}(\mathbf{H}^{(n-1)})^T-\mathbf{I}\right\|^2+\frac{\lambda_4}{2m}\left\|\mathbf{H}^{(n-1)}\mathbf{1}_{m\times 1}\right\|^2$$

$$\text{s.t. } \mathbf{B}\in\{-1,1\}^{L\times m}$$

- 1^{st} term: reconstruction error → encourage neighbor preserving
- 3^{rd} term: binary quantization loss
- 4^{th} and 5^{th} terms: independence and balance of codes

Supervised Hashing



$$\min_{\mathbf{W},\mathbf{c},\mathbf{B}} \frac{1}{2m}\left\|\frac{1}{L}(\mathbf{H}^{(n)})^T\mathbf{H}^{(n)}-\mathbf{S}\right\|^2+\frac{\lambda_1}{2}\sum_{l=1}^{n-1}\left\|\mathbf{W}^{(l)}\right\|^2$$

$$+\frac{\lambda_2}{2m}\left\|\mathbf{H}^{(n)}-\mathbf{B}\right\|^2+\frac{\lambda_3}{2}\left\|\frac{1}{m}\mathbf{H}^{(n)}(\mathbf{H}^{(n)})^T-\mathbf{I}\right\|^2+\frac{\lambda_4}{2m}\left\|\mathbf{H}^{(n)}\mathbf{1}_{m\times 1}\right\|^2$$

$$\text{s.t. } \mathbf{B}\in\{-1,1\}^{L\times m}$$

where $\mathbf{S}_{ij}=1$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are same class; -1 otherwise

Optimization

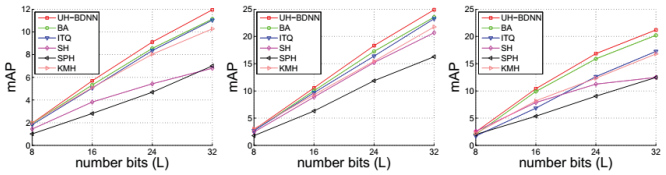
Alternative optimize over network weight $(\mathbf{W},\mathbf{c})$ and $\mathbf{B}$.

- When fixing $\mathbf{B}$ → unconstrained opt w.r.t. $(\mathbf{W},\mathbf{c})$: solving by $LBFGS$
- When fixing $(\mathbf{W},\mathbf{c})$ → binary opt w.r.t. $\mathbf{B}$
  - Unsupervised: coordinate descent, i.e., solve one row of $\mathbf{B}$ at a time → closed form for that row. Let $\mathbf{V}=\mathbf{X}-\mathbf{c}^{(n-1)}\mathbf{1}_{1\times m}$; $\mathbf{Q}=(\mathbf{W}^{(n-1)})^T\mathbf{V}+\lambda_2\mathbf{H}^{(n-1)}$. For $k=1,\cdots L$, let $\mathbf{w}_k$ be k^{th} column of $\mathbf{W}^{(n-1)}$; $\mathbf{W}_1$ be matrix $\mathbf{W}^{(n-1)}$ excluding $\mathbf{w}_k$; $\mathbf{q}_k$ be k^{th} column of $\mathbf{Q}^T$; $\mathbf{b}_k^T$ be k^{th} row of $\mathbf{B}$; $\mathbf{B}_1$ be matrix of $\mathbf{B}$ excluding $\mathbf{b}_k^T$.
    We have closed form for $\mathbf{b}_k^T=sgn(\mathbf{q}_k^T-\mathbf{w}_k^T\mathbf{W}_1\mathbf{B}_1)$
  - Supervised: closed form $\mathbf{B}=sgn(\mathbf{H}^{(n)})$

Dataset

- CIFAR10: 60K images; 10K testing; AlexNet features
- MNIST: 70K images; 10K testing; raw (intensity) features
- SIFT1M: 1M features; 10K testing; SIFT features

Evaluation of Unsupervised Hashing



(a) CIFAR10 (b) MNIST (c) SIFT1M

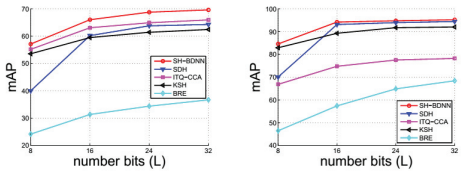
Figure 1: mAP comparison between UH-BDNN and the state of the art.

	CIFAR10				MNIST				SIFT1M			
L	8	16	24	32	8	16	24	32	8	16	24	32
UH-BDNN	0.55	5.79	22.14	18.35	0.53	6.80	29.38	38.50	4.80	25.20	62.20	80.55
BA	0.55	5.65	20.23	17.00	0.51	6.44	27.65	35.29	3.85	23.19	61.35	77.15
ITQ	0.54	5.05	18.82	17.76	0.51	5.87	23.92	36.35	3.19	14.07	35.80	58.69
SH	0.39	4.23	14.60	15.22	0.43	6.50	27.08	36.69	4.67	24.82	60.25	72.40
SPH	0.43	3.45	13.47	13.67	0.44	5.02	22.24	30.80	4.25	20.98	47.09	66.42
KMH	0.53	5.49	19.55	15.90	0.50	6.36	25.68	36.24	3.74	20.74	48.86	76.04

Table 1: Precision at Hamming distance $r=2$ comparison between UH-BDNN and the state of the art.

- UH-BDNN is comparable or outperforms other methods
- The higher code length, the more improvement

Evaluation of Supervised Hashing



(a) CIFAR10 (b) MNIST

Figure 2: mAP comparison between SH-BDNN and the state of the art.

	CIFAR10				MNIST			
L	8	16	24	32	8	16	24	32
SH-BDNN	54.12	67.32	69.36	69.62	84.26	94.67	94.69	95.51
SDH	31.60	62.23	67.65	67.63	36.49	93.00	93.98	94.43
ITQ-CCA	49.14	65.68	67.47	67.19	54.35	79.99	84.12	84.57
KSH	44.81	64.08	67.01	65.76	68.07	90.79	92.86	92.41
BRE	23.84	41.11	47.98	44.89	37.67	69.80	83.24	84.61

Table 2: Precision at Hamming distance $r=2$ comparison between SH-BDNN and the state of the art.

- SH-BDNN is competitive or outperforms compared methods

Comparison with CNN-based hashing methods:

- CIFAR10: 50K training; 10K testing (with leave-one-out procedure)

	mAP				precison@2			
L	16	24	32	48	16	24	32	48
SH-BDNN	64.30	65.21	66.22	66.53	56.87	58.67	58.80	58.42
DRSCH[33]	61.46	62.19	62.87	63.05	52.34	53.07	52.31	52.03
DSRH[32]	60.84	61.08	61.74	61.77	50.36	52.45	50.37	49.38

Table 3: Comparison between SH-BDNN and CNN-based hashing DSRH [CVPR'15] and DRSCH [TIP'15] on CIFAR10

- SH-BDNN achieves more than 3% improvement over the state of the art.

Conclusion

- Unified framework based on deep neural network for both unsupervised and supervised hashing
- Directly produce binary code at one layer
- Ensure good properties of codes: similarity preserving, independence, balance.
- Code release: `http://tinyurl.com/hx4f44f`