HashNet: Deep Learning to Hash by Continuation

Zhangjie Cao†, Mingsheng Long†, Jianmin Wang†, and Philip S. Yu†‡

†KLiss, MOE; NEL-BDS; TNList; School of Software, Tsinghua University, China
‡University of Illinois at Chicago, IL, USA

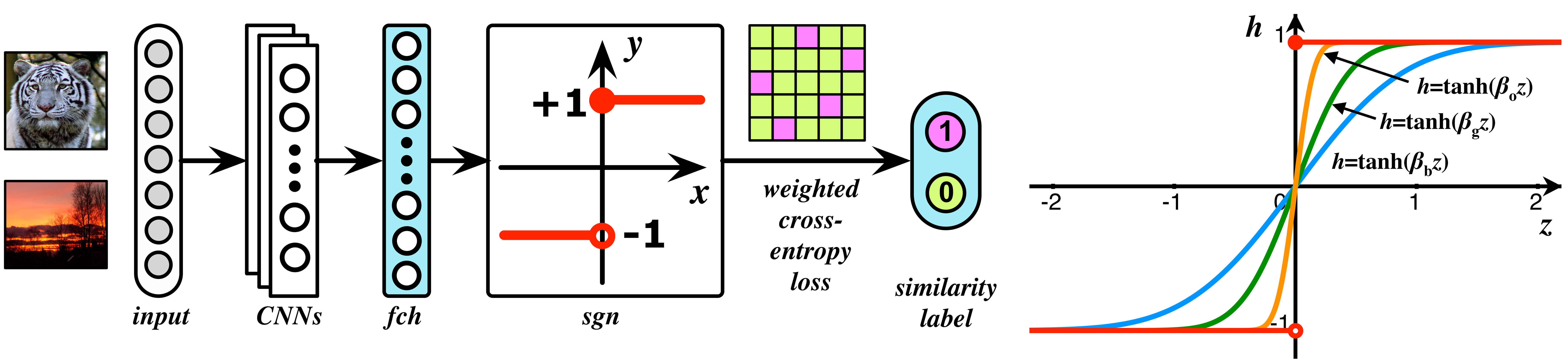


Figure 1: (left) The proposed HashNet for deep learning to hash by continuation, which is comprised of four key components: (1) Standard convolutional neural network (CNN), e.g. AlexNet and ResNet, for learning deep image representations, (2) a fully-connected hash layer (fch) for transforming the deep representation into K-dimensional representation, (3) a sign activation function (sgn) for binarizing the K-dimensional representation into K-bit binary hash code, and (4) a novel weighted cross-entropy loss for similarity-preserving learning from sparse data. (right) Plot of smoothed responses of the sign function $h = \mathrm{sgn}(z)$: Red is the sign function, and blue, green and orange show functions $h = \tanh(\beta z)$ with bandwidths $\beta_b < \beta_g < \beta_o$. The key property is $\lim_{\beta\to\infty} \tanh(\beta z) = \mathrm{sgn}(z)$. *Best viewed in color.*

Summary

1. An end-to-end deep learning to hash architecture for image retrieval which unifies four key components:
- a standard convolutional neural network
- a fully-connected hash layer (fch) for transforming the deep representation into K-dimensional representation
- a sign activation function (sgn) for binarizing the K-dimensional representation into K-bit binary hash code
- a novel weighted cross-entropy loss for similarity-preserving learning from sparse data

2. Learning by continuation method to optimize HashNet

Challenges

1. Limitations of previous works:
- *Ill-posed gradient* problem: We need the ***sign*** function as activation function to convert deep *continuous* representations to *binary* hash codes. But the gradient of the sign function is zero for all nonzero inputs, making standard back-propagation infeasible.
  → attack the *ill-posed gradient* problem by the *continuation* methods, which address a complex optimization problem by smoothing the original function, turning it into a easier to optimize problem.
- *Data imbalance* problem: The number of similar pairs is much smaller than that of dissimilar pairs in real retrieval systems which makes similarity-preserving learning ineffective.
  → propose a novel weighted cross-entropy loss for similarity-preserving learning from sparse similarity.

Model Formulation

Weighted Maximum Likelihood (WML) estimation:

$$\log P(\mathcal{S}|\boldsymbol{H}) = \sum_{s_{ij}\in\mathcal{S}} w_{ij} \log P(s_{ij}|\boldsymbol{h}_i, \boldsymbol{h}_j), \tag{1}$$

Weight Formulation:

$$w_{ij} = c_{ij} \cdot \begin{cases} |\mathcal{S}| \,/\, |\mathcal{S}_1|, & s_{ij} = 1 \\ |\mathcal{S}| \,/\, |\mathcal{S}_0|, & s_{ij} = 0 \end{cases} \tag{2}$$

Pairwise logistic function:

$$\begin{aligned} P(s_{ij}|\boldsymbol{h}_i, \boldsymbol{h}_j) &= \begin{cases} \sigma(\langle \boldsymbol{h}_i, \boldsymbol{h}_j \rangle), & s_{ij} = 1 \\ 1 - \sigma(\langle \boldsymbol{h}_i, \boldsymbol{h}_j \rangle), & s_{ij} = 0 \end{cases} \\ &= \sigma(\langle \boldsymbol{h}_i, \boldsymbol{h}_j \rangle)^{s_{ij}} (1 - \sigma(\langle \boldsymbol{h}_i, \boldsymbol{h}_j \rangle))^{1-s_{ij}} \end{aligned} \tag{3}$$

Optimization problem of HashNet:

$$\min_{\Theta} \sum_{s_{ij}\in\mathcal{S}} w_{ij} \left(\log\left(1 + \exp\left(\alpha \langle \boldsymbol{h}_i, \boldsymbol{h}_j \rangle\right)\right) - \alpha s_{ij} \langle \boldsymbol{h}_i, \boldsymbol{h}_j \rangle \right), \tag{4}$$

Learning by Continuation

Algorithm:
Input: A sequence $1 = \beta_0 < \beta_1 < \ldots < \beta_m = \infty$
for *stage* $t = 0$ ***to*** m **do**
  Train HashNet (4) with $\tanh(\beta_t z)$ as activation Set converged HashNet as next stage initialization
end
Output: HashNet with $\mathrm{sgn}(z)$ as activation, $\beta_m \to \infty$

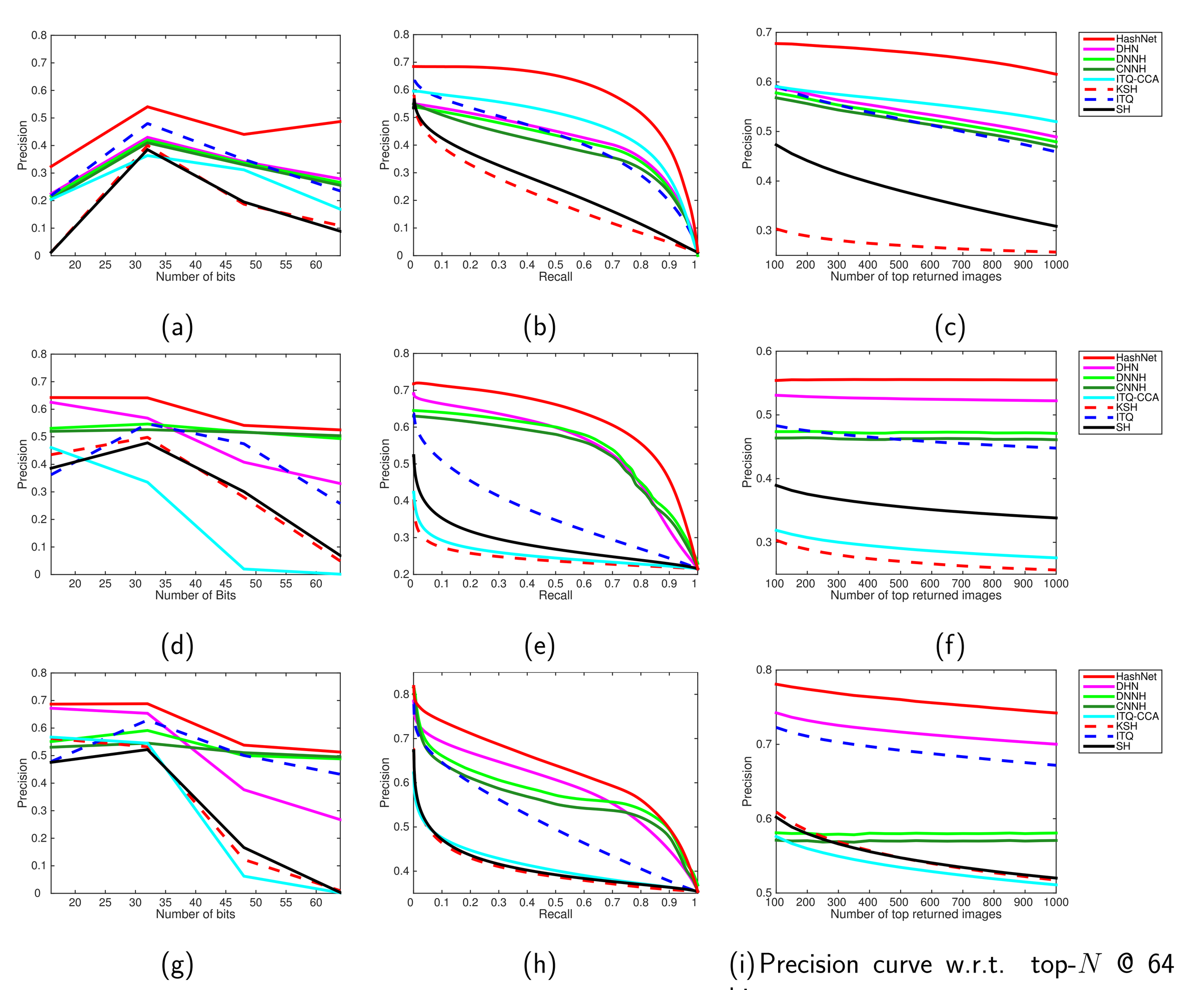
Experiment Setup

Datasets: ImageNet, NUS-WIDE and MS COCO
Baselines: LSH, SH, ITQ, ITQ-CCA, BRE, KSH, SDH, CNNH, DNNH and DHN
Criteria: MAP, Precision-recall curve and Precision@top R curve

Results

Dataset	Method	MAP 16 bits	MAP 32 bits	MAP 48 bits	MAP 64 bits
ImageNet	HashNet	**0.5059**	**0.6306**	**0.6633**	**0.6835**
	DHN	0.3106	0.4717	0.5419	0.5732
	DNNH	0.2903	0.4605	0.5301	0.5645
	CNNH	2812	0.4498	0.5245	0.5538
	SDH	0.2985	0.4551	0.5549	0.5852
	KSH	0.1599	0.2976	0.3422	0.3943
	ITQ-CCA	0.2659	0.4362	0.5479	0.5764
	ITQ	0.3255	0.4620	0.5170	0.5520
	BRE	0.5027	0.5290	0.5475	0.5546
	SH	0.2066	0.3280	0.3951	0.4191
	LSH	0.1007	0.2350	0.3121	0.3596
NUS-WIDE	HashNet	**0.6623**	**0.6988**	**0.7114**	**0.7163**
	DHN	0.6374	0.6637	0.6692	0.6714
	DNNH	0.5976	0.6158	0.6345	0.6388
	CNNH	0.5696	0.5827	0.5926	0.5996
	SDH	0.4756	0.5545	0.5786	0.5812
	KSH	0.3561	0.3327	0.3124	0.3368
	ITQ-CCA	0.4598	0.4052	0.3732	0.3467
	ITQ	0.5086	0.5425	0.5580	0.5611
	BRE	0.0628	0.2525	0.3300	0.3578
	SH	0.4058	0.4209	0.4211	0.4104
	LSH	0.3283	0.4227	0.4333	0.5009
MS COCO	HashNet	**0.6873**	**0.7184**	**0.7301**	**0.7362**
	DHN	0.6774	0.7013	0.6948	0.6944
	DNNH	0.5932	0.6034	0.6045	0.6099
	CNNH	0.5642	0.5744	0.5711	0.5671
	SDH	0.5545	0.5642	0.5723	0.5799
	KSH	0.5212	0.5343	0.5343	0.5361
	ITQ-CCA	0.5659	0.5624	0.5297	0.5019
	ITQ	0.5818	0.6243	0.6460	0.6574
	BRE	0.5920	0.6224	0.6300	0.6336
	SH	0.4951	0.5071	0.5099	0.5101
	LSH	0.4592	0.4856	0.5440	0.5849

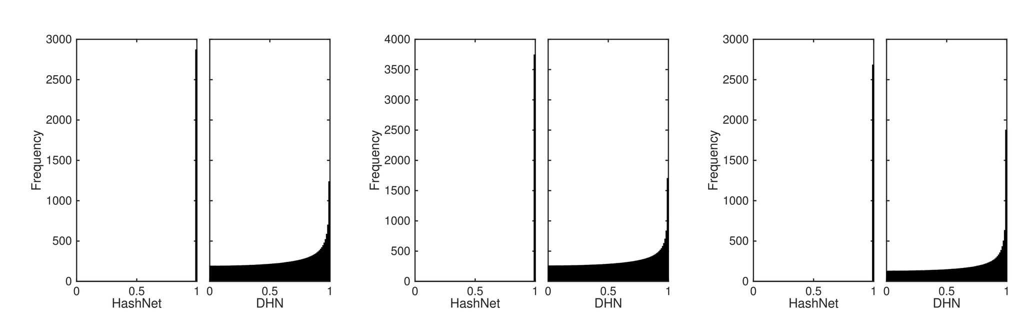
Table 1: MAP of Hamming Ranking on the three datasets



(a) (b) (c) (d) (e) (f) (g) (h) (i) Precision curve w.r.t. top-N @ 64 bits

Figure 2: The experimental results of Precision within Hamming radius 2, Precision-recall curve @ 64 bits and Precision curve w.r.t. top-N @ 64 bits on the three datasets.

Code Quality



(a) ImageNet (b) NUS-WIDE (c) COCO

Figure 3: Histogram of non-binarized codes of HashNet and DHN.

Ablation Study

Method	ImageNet 16 bits	ImageNet 32 bits	ImageNet 48 bits	ImageNet 64 bits	NUS-WIDE 16 bits	NUS-WIDE 32 bits	NUS-WIDE 48 bits	NUS-WIDE 64 bits
HashNet+C	**0.5059**	**0.6306**	**0.6633**	**0.6835**	**0.6646**	**0.7024**	**0.7209**	**0.7259**
HashNet	**0.5059**	**0.6306**	**0.6633**	**0.6835**	0.6623	0.6988	0.7114	0.7163
HashNet-W	0.3350	0.4852	0.5668	0.5992	0.6400	0.6638	0.6788	0.6933
HashNet-sgn	0.4249	0.5450	0.5828	0.6061	0.6603	0.6770	0.6921	0.7020

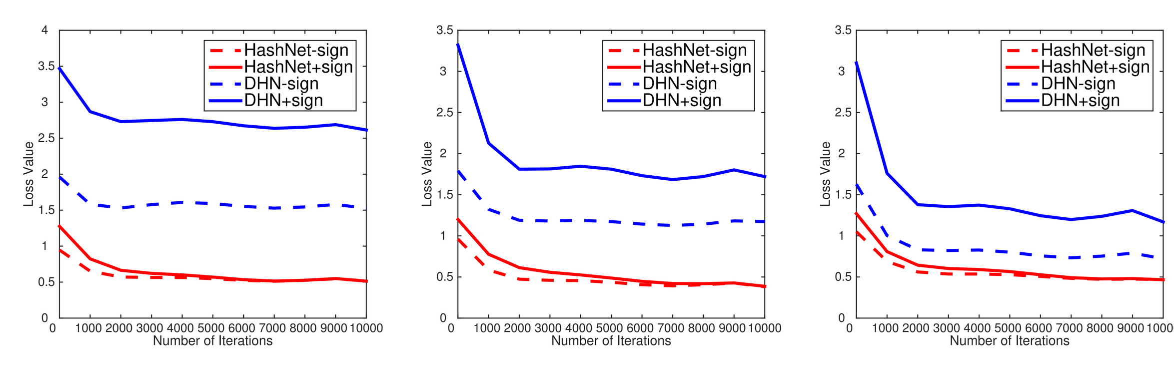
Table 2: MAP Results of HashNet and Its Variants on ImageNet and NUS-WIDE

Table 3: MAP Results of HashNet and Its Variants on MS COCO

Method	MS COCO 16 bits	MS COCO 32 bits	MS COCO 48 bits	MS COCO 64 bits
HashNet+C	**0.6876**	**0.7261**	**0.7371**	**0.7419**
HashNet	0.6873	0.7184	0.7301	0.7362
HashNet-W	0.6853	0.7174	0.7297	0.7348
HashNet-sgn	0.6449	0.6891	0.7056	0.7138

HashNet+C: variant using continuous similarity.
HashNet-W: variant using maximum likelihood without weight.
HashNet-sgn: variant using tanh() not sgn() as activation function.

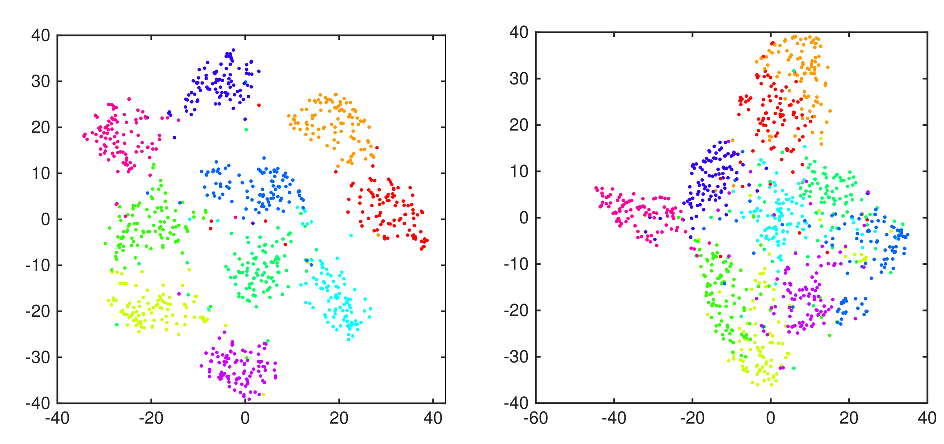
Convergence Analysis



(a) ImageNet (b) NUS-WIDE (c) COCO

Figure 4: Losses of HashNet and DHN through training process.

Visualization



(a) HashNet (b) DHN

Figure 5: The t-SNE of hash codes learned by HashNet and DHN.

Contact Information

- Web: `http://ise.thss.tsinghua.edu.cn/~mlong/`
- Email: `mingsheng@tsinghua.edu.cn`