Supplemental Material:
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
caozhangjie14@gmail.com {mingsheng,jimwang}@tsinghua.edu.cn psyu@uic.edu

1. Convergence Analysis

We briefly analyze that the continuation optimization in Algorithm 1 will decrease the loss of HashNet (4) in each stage and in each iteration until converging to HashNet with sign activation function that generates *exactly* binary codes.

Let $L_{ij} = w_{ij} \left(\log\left(1+\exp\left(\alpha \left\langle \boldsymbol{h}_i, \boldsymbol{h}_j \right\rangle\right)\right) - \alpha s_{ij} \left\langle \boldsymbol{h}_i, \boldsymbol{h}_j \right\rangle\right)$ and $L = \sum_{s_{ij} \in \mathcal{S}} L_{ij}$, where $\boldsymbol{h}_i \in \{-1,+1\}^K$ are *binary* hash codes. Note that when optimizing HashNet by continuation in Algorithm 1, network activation in each stage t is $g = \tanh(\beta_t z)$, which is *continuous* in nature and will only become *binary* when convergence $\beta_t \to \infty$. Denote by $J_{ij} = w_{ij} \left(\log\left(1+\exp\left(\alpha \left\langle \boldsymbol{g}_i, \boldsymbol{g}_j \right\rangle\right)\right) - \alpha s_{ij} \left\langle \boldsymbol{g}_i, \boldsymbol{g}_j \right\rangle\right)$ and $J = \sum_{s_{ij} \in \mathcal{S}} J_{ij}$ the true loss we optimize in Algorithm 1, where $\boldsymbol{g}_i \in \mathbb{R}^K$ and note that $\boldsymbol{h}_i = \operatorname{sgn}(\boldsymbol{g}_i)$. We will show that HashNet loss $L(\boldsymbol{h})$ descends when minimizing $J(\boldsymbol{g})$.

Theorem 1. *The HashNet loss L will not change across stages t and $t+1$ with bandwidths switched from β_t to β_{t+1}.*

Proof. When the algorithm switches from stages t to $t+1$ with bandwidths changed from β_t to β_{t+1}, only the network activation is changed from $\tanh(\beta_t z)$ to $\tanh(\beta_{t+1} z)$ but its sign $h = \operatorname{sgn}(\tanh(\beta_t z)) = \operatorname{sgn}(\tanh(\beta_{t+1} z))$, i.e. the hash code, remains the same. Thus L is unchanged. $\square$

For each pair of binary codes $\boldsymbol{h}_i, \boldsymbol{h}_j$ and their continuous counterparts $\boldsymbol{g}_i, \boldsymbol{g}_j$, the derivative of J w.r.t. each bit k is

$$\frac{\partial J}{\partial g_{ik}} = w_{ij} \alpha \left(\frac{1}{1+\exp\left(-\alpha \left\langle \boldsymbol{g}_i, \boldsymbol{g}_j \right\rangle\right)} - s_{ij} \right) g_{jk}, \tag{1}$$

where $k = 1, \ldots, K$. The derivative of J w.r.t. $\boldsymbol{g}_j$ can be defined similarly. Updating $\boldsymbol{g}_i$ by SGD, the updated $\boldsymbol{g}_i'$ is

$$\begin{aligned} g_{ik}' &= g_{ik} - \eta \frac{\partial J}{\partial g_{ik}} \\ &= g_{ik} - \eta w_{ij} \alpha \left(\frac{1}{1+\exp\left(-\alpha \left\langle \boldsymbol{g}_i, \boldsymbol{g}_j \right\rangle\right)} - s_{ij} \right) g_{jk}, \end{aligned} \tag{2}$$

where η is the learning rate and $\boldsymbol{g}_j'$ is computed similarly.

Lemma 1. *Denote by $\boldsymbol{h}_i = \operatorname{sgn}(\boldsymbol{g}_i)$, $\boldsymbol{h}_i' = \operatorname{sgn}(\boldsymbol{g}_i')$, then*

$$\begin{cases} \left\langle \boldsymbol{h}_i', \boldsymbol{h}_j' \right\rangle \geqslant \left\langle \boldsymbol{h}_i, \boldsymbol{h}_j \right\rangle, & s_{ij} = 1, \\ \left\langle \boldsymbol{h}_i', \boldsymbol{h}_j' \right\rangle \leqslant \left\langle \boldsymbol{h}_i, \boldsymbol{h}_j \right\rangle, & s_{ij} = 0. \end{cases} \tag{3}$$

Proof. Since $\left\langle \boldsymbol{h}_i, \boldsymbol{h}_j \right\rangle = \sum_{k=1}^K h_{ik} h_{jk}$, Lemma 1 can be proved by verifying that $h_{ik}' h_{jk}' \geqslant h_{ik} h_{jk}$ if $s_{ij} = 1$ and $h_{ik}' h_{jk}' \leqslant h_{ik} h_{jk}$ if $s_{ij} = 0$, $\forall k = 1, 2, \ldots, K$.

Case 1. $s_{ij} = 0$.

(1) If $g_{ik} < 0$, $g_{jk} > 0$, then $\frac{\partial J}{\partial g_{ik}} > 0$, $\frac{\partial J}{\partial g_{jk}} < 0$. Thus, $h_{ik}' \leqslant h_{ik} = -1$, $h_{jk}' \geqslant h_{jk} = 1$. And we have $h_{ik}' h_{jk}' = -1 = h_{ik} h_{jk}$.

(2) If $g_{ik} > 0$, $g_{jk} < 0$, then $\frac{\partial J}{\partial g_{ik}} < 0$, $\frac{\partial J}{\partial g_{jk}} > 0$. Thus, $h_{ik}' \geqslant h_{ik} = 1$, $h_{jk}' \leqslant h_{jk} = -1$. And we have $h_{ik}' h_{jk}' = -1 = h_{ik} h_{jk}$.

(3) If $g_{ik} < 0$, $g_{jk} < 0$, then $\frac{\partial J}{\partial g_{ik}} < 0$, $\frac{\partial J}{\partial g_{jk}} < 0$. Thus $h_{ik}' \geqslant h_{ik} = -1$, $h_{jk}' \geqslant h_{jk} = -1$. So h_{ik}' and h_{jk}' may be either $+1$ or -1 and we have $h_{ik}' h_{jk}' \leqslant 1 = h_{ik} h_{jk}$.

(4) If $g_{ik} > 0$, $g_{jk} > 0$, then $\frac{\partial J}{\partial g_{ik}} > 0$, $\frac{\partial J}{\partial g_{jk}} > 0$. Thus $h_{ik}' \leqslant h_{ik} = 1$, $h_{jk}' \leqslant h_{jk} = 1$. So h_{ik}' and h_{jk}' may be either $+1$ or -1 and we have $h_{ik}' h_{jk}' \leqslant 1 = h_{ik} h_{jk}$.

Case 2. $s_{ij} = 1$. *It can be proved similarly as Case 1.* $\square$

Theorem 2. *Loss L decreases when optimizing loss $J(\boldsymbol{g})$ by the stochastic gradient descent (SGD) within each stage.*

Proof. The gradient of loss L w.r.t. *hash* codes $\left\langle \boldsymbol{h}_i, \boldsymbol{h}_j \right\rangle$ is

$$\frac{\partial L}{\partial \left\langle \boldsymbol{h}_i, \boldsymbol{h}_j \right\rangle} = w_{ij} \alpha \left(\frac{1}{1+\exp\left(-\alpha \left\langle \boldsymbol{h}_i, \boldsymbol{h}_j \right\rangle\right)} - s_{ij} \right). \tag{4}$$

We observe that

$$\begin{cases} \frac{\partial L}{\partial \left\langle \boldsymbol{h}_i, \boldsymbol{h}_j \right\rangle} \leqslant 0, & s_{ij} = 1, \\ \frac{\partial L}{\partial \left\langle \boldsymbol{h}_i, \boldsymbol{h}_j \right\rangle} \geqslant 0, & s_{ij} = 0. \end{cases} \tag{5}$$

By substituting Lemma 1: if $s_{ij} = 1$, then $\left\langle \boldsymbol{h}_i', \boldsymbol{h}_j' \right\rangle \geqslant \left\langle \boldsymbol{h}_i, \boldsymbol{h}_j \right\rangle$, and thus $L(\boldsymbol{h}_i', \boldsymbol{h}_j') \leqslant L(\boldsymbol{h}_i, \boldsymbol{h}_j)$; if $s_{ij} = 0$, then $\left\langle \boldsymbol{h}_i', \boldsymbol{h}_j' \right\rangle \leqslant \left\langle \boldsymbol{h}_i, \boldsymbol{h}_j \right\rangle$, and thus $L(\boldsymbol{h}_i', \boldsymbol{h}_j') \leqslant L(\boldsymbol{h}_i, \boldsymbol{h}_j)$. $\square$