Deep Quantization Network for Efficient Image Retrieval

Yue Cao, Mingsheng Long, Jianmin Wang, Han Zhu, and Qingfu Wen

School of Software
Tsinghua University

The Thirtieth AAAI Conference on Artificial Intelligence, 2016
Large Scale Image Retrieval

Find Visually Similar Images

From these Images
Hashing Methods

**Superiorities**

**Memory**
- 128-d float: 512 bytes → 16 bytes
- 1 billion items: 512 GB → 16 GB

**Time**
- Computation: ×10 - ×100 faster
- Transmission (disk / web): ×30 faster

**Categories**
- Hamming Embedding Methods
- Quantization Methods

**Applications**
- Approximate nearest neighbor search
- Compact representation, Feature Compression for large datasets
Vector Quantization

- # code words: $K$
- code length: $B = \log_2 K$

- $x \rightarrow c_i$ (nearest codeword)
- code stored: $i(x)$
Vector Quantization

Vector Quantization

- # code words: $K$
- code length: $B = \log_2 K$

VQ for ANN Search

$$d(x, y) \approx d(c_i, c_j) \triangleq \text{lookup}(i, j)$$

construct a $K$-by-$K$ (also $2^B$-by-$2^B$) look-up table
Product Quantization (PQ) [pami 11']

Loss

\[
\min_{c_1, \ldots, c_M} \sum_{i=1}^{N} \left\| x_i - c_i(x) \right\|^2 \\
\text{s.t. } c \in c_1 \times c_2 \times \ldots \times c_M
\]

Pros

- Huge codebook: \( K = k^M \)
- Tractable: \( M k \)-by-\( k \) tables

Cons

- Sensitive to Projection
Optimized Product Quantization (OPQ) [cvpr 13']

Loss

\[
\min_{R, c_1, \ldots, c_M} \sum_{i=1}^{N} \| x_i - c_{i(x)} \|^2
\]

s.t. \( R \in c_1 \times c_2 \times \ldots \times c_M \),
\( R^T R = I \)

Pros

- Huge codebook: \( K = k^M \)
- Tractable: \( M \) \( k \)-by-\( k \) tables
- Insensitive for rotation

Cons

- high correlated between subspaces
OPQ with Deep Features

Pros

- Insensitive for rotation
- Low correlated between subspaces

Cons

- Poor quantizability: Input vector cannot be easily clustered into clusters.

Y. Cao et al. (Tsinghua University)
Deep Quantization

Product Quantization Loss

\[ Q = \sum_{i=1}^{N} \| z_i^l - Ch_i \|_2^2, \quad (1) \]

\[ C = \text{diag}(C_1, C_2, \ldots, C_M) = \begin{bmatrix} C_1 & 0 & \cdots & 0 \\ 0 & C_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_M \end{bmatrix}. \]

Pros

- low correlated between subspaces
- easily clustered for each subspace (high quantizability)
- Look-up table is the same as PQ
Motivation

Quantization Methods

Similarity Preserving

Previous works [cvpr12', aaai14']

\[ L = \sum_{s_{ij} \in S} \left( s_{ij} - \frac{1}{B} \langle z_i, z_j \rangle \right)^2 \]

\( \langle z_i, z_j \rangle \in [-R, R] \) but \( s_{ij} \in \{-1, 1\} \).

Our approach

\[ L = \sum_{s_{ij} \in S} \left( s_{ij} - \frac{\langle z_i, z_j \rangle}{\|z_i\| \|z_j\|} \right)^2 \]

\( \cos(z_i, z_j) = \langle z_i, z_j \rangle / \|z_i\| \|z_j\| \in [-1, 1] \) with \( s_{ij} \in \{-1, 1\} \), hence making our loss well-specified for preserving the similarity conveyed in \( S \).
Objective Function

\[
\min_{\Theta, C, H} L + \lambda Q, \quad (2)
\]

Pairwise Cosine Loss

\[
L = \sum_{s_{ij} \in S} \left( s_{ij} - \frac{\langle z_i^l, z_j^l \rangle}{\|z_i^l\| \|z_j^l\|} \right)^2, \quad (3)
\]

Product Quantization Loss

\[
Q = \sum_{i=1}^{N} \|z_i^l - Ch_i\|_2^2, \quad (4)
\]

\[
C = \text{diag} \left( C_1, C_2, \ldots, C_M \right) = \begin{bmatrix}
C_1 & 0 & \cdots & 0 \\
0 & C_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & C_M
\end{bmatrix}.
\]
Deep Quantization Network

Key Contributions

- An end-to-end deep quantization framework using Alexnet for deep representation learning
- Firstly minimize quantization error with deep representation learning, which significantly improve quantizability
- Devise a pairwise cosine loss to better link the cosine distances with similarity labels
Approximate Nearest Neighbor Search

Asymmetric Quantizer Distance (AQD)

\[
AQD(q, x_i) = \sum_{m=1}^{M} \left\| z_{qm} - C_m h_{im} \right\|_2^2 \tag{5}
\]

- \(q\): query
- \(x_i\): raw feature of db point \(i\)
- \(z_q^l\): deep representation of query \(q\)
- \(h_{im}\): binary code of \(x_i\) in \(m\)-th subspace
- \(C_m h_{im}\): compressed representation of \(x_i\) in \(m\)-th subspace

Look-up Tables

- For each query, pre-compute \(M \times K\) Look-up table
- Each query entails \(M\) table lookups and additions
Theoretical Analysis

Theorem (Error Bound)

The error of using AQD (5) to approximate original Euclidean distance is bounded by the product quantization error (4)

\[ |AQB (q, x_i) - d (q, x_i)| \leq \left\| z_q^i - Ch_i \right\|_2 + |\epsilon|. \]  

(6)

where \( d (q, x_i) = \left\| z_q^i - z_i^l \right\|_2 \).

The theorem can be easily proved by triangle inequality.

Insights

The error of using AQD is statistically bounded by DQN quantization loss (4), which indicates that DQN is more accurate than sign thresholding methods which do not control the quantization error.
Experiment Setup

- **Datasets:** pre-trained on ImageNet, finetuned on Nus-wide, Cifar-10 and MIRFlickr25k
- **Protocols:** MAPs, Precision-Recall Curve, Precision Top-R Curve
- **Parameter selection:** cross-validation by jointly assessing
  - test errors of joint loss function

(Fei-Fei et al. 2012) → Pre-train → Caffe → Fine-tune → CIFAR-10

(Jia et al. 2014) → Y. Cao et al. (Tsinghua University)

(Krizhevsky et al. 2009)
Results and Discussion

Learning Hash Codes by end-to-end deep hashing approach

- Product Quantization, Pairwise Cosine Loss with Alexnet (DQN) vs. Triplet Deep Hash with NiN structure (DNNH) vs. Best Shallow Hash with deep fc7 features (KSH-D)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NUS-WIDE</th>
<th>CIFAR-10</th>
<th>Flickr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12 bits</td>
<td>24 bits</td>
<td>32 bits</td>
</tr>
<tr>
<td>KSH</td>
<td>0.556</td>
<td>0.572</td>
<td>0.581</td>
</tr>
<tr>
<td>KSH-D</td>
<td>0.673</td>
<td>0.705</td>
<td>0.717</td>
</tr>
<tr>
<td>CNNH</td>
<td>0.617</td>
<td>0.663</td>
<td>0.657</td>
</tr>
<tr>
<td>DNNH</td>
<td>0.674</td>
<td>0.697</td>
<td>0.713</td>
</tr>
<tr>
<td>DQN</td>
<td><strong>0.768</strong></td>
<td><strong>0.776</strong></td>
<td><strong>0.783</strong></td>
</tr>
</tbody>
</table>

(a) NUS-WIDE  (b) CIFAR-10  (c) NUS-WIDE  (d) CIFAR-10
Empirical Analysis

\[ DQN_{2+}: \text{two step method, Similarity Preserving + OPQ} \]
\[ DQN_{ip}: \text{replace pairwise cosine loss with Inner-Product loss} \]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NUS-WIDE</th>
<th>CIFAR-10</th>
<th>Flickr</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQN_{2+}</td>
<td>0.750 0.754 0.756 0.764</td>
<td>0.528 0.534 0.538 0.541</td>
<td>0.804 0.809 0.815 0.829</td>
</tr>
<tr>
<td>DQN_{ip}</td>
<td>0.623 0.646 0.655 0.673</td>
<td>0.506 0.513 0.519 0.529</td>
<td>0.748 0.756 0.759 0.775</td>
</tr>
<tr>
<td>DQN</td>
<td><strong>0.768 0.776 0.783 0.792</strong></td>
<td><strong>0.554 0.558 0.564 0.580</strong></td>
<td><strong>0.839 0.848 0.854 0.863</strong></td>
</tr>
</tbody>
</table>

**Key Observations**

- DQN outperforms DQN_{2+}, indicating product quantization error with representation learning can **boost the quantizability**.
- DQN outperforms DQN_{ip} by large margins, indicating the **superiority** of cosine similarity and the **inconsistency** of inner-product loss.
Summary

- A deep quantization network (DQN) for efficient image retrieval
- Three important contributions
  - An end-to-end deep quantization framework using Alexnet for deep representation learning
  - Firstly minimize quantization error with deep representation learning, which significantly improve quantizability
  - Devise a pairwise cosine loss to better link the cosine distances with similarity labels

- Open Problems
  - Inverted multi-index for Deep Quantization Networks
  - Deeper convolutional neural networks for better representation learning