Inductive Transfer Deep Hashing for Image Retrieval

Xinyu Ou
Huazhong University of Science and Technology
1037 Luoyu Rd., Wuhan 430074, China
Yunnan Open University
113 Xuefu Rd., Kunming 650223, China
ouxinyu@hust.edu.cn

Lingyu Yan, Hefei Ling*, Cong Liu,
Maolin Liu
Huazhong University of Science and Technology
1037 Luoyu Rd., Wuhan 430074, China
{yanranyaya, lhefei}@hust.edu.cn,
luccongwust@gmail.com,
u201014564@hust.edu.cn

ABSTRACT

With the explosive increase of online images, fast similarity search is increasingly critical for large scale image retrieval. Several hashing methods have been proposed to accelerate image retrieval, a promising way is semantic hashing which designs compact binary codes for a large number of images so that semantically similar images are mapped to similar codes. Supervised methods can handle such semantic similarity but they are prone to overfitting when the labeled data is few or noisy. In this paper, we concentrate on this issue and propose a novel Inductive Transfer Deep Hashing (ITDH) approach for semantic hashing based image retrieval. A transfer deep learning algorithm has been employed to learn the robust image representation, and the neighborhood-structure preserved method has been used to mapped the image into discriminative hash codes in hashing space. The combination of the two techniques ensures that we obtain a good feature representation and a fast query speed without depending on large amounts of labeled data. Experimental results demonstrate that the proposed approach is superior to some state-of-the-art methods.

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval]: Retrieval models, Search process

Keywords
Inductive Transfer Learning; Deep Learning; Image Retrieval; Neighborhood-Structure Preserved; Semantic Hashing

1. INTRODUCTION

Due to the exponential growth of online images, it is necessary to consider searching among large number of images, which is very time-consuming and unscaleable. To tackle the problem of large scale image applications, machine learning is introduced into the field of image representation to discover the hidden semantic patterns among images.

Based on feature extraction, many hashing methods are proposed to generate compact and discriminative image representation. Tang et al. [11] develop a global method using nonnegative matrix factorization (NMF), which first convert image into a fixed-sized pixel array and then generate secondary image by rearranging pixels and applying NMF to produce a feature-bearing hash code, after that, the fingerprint is coarsely quantized into binary string and scrambled to generate the image fingerprint. Wang et al. [12] have proposed a semi-supervised hashing method (SSH) that incorporates pairwise semantic similarity and dissimilarity constraints from labeled data, which minimized the empirical error on the labeled data while maximizing entropy of the generated hash bits over the unlabeled data. Similarity Sensitive Coding (SSC) [9] adopts boosting approach, they first train AdaBoost classifiers with similar pairs of items as positive examples (and also non-similar pairs of items as negative examples in SCC), and then take the output of all (decision stump) weak learners on a given document as its binary code. Self-taught hashing (STH) [14] is proposed and considered as one of the state-of-the-art works [6]. However, it suffers overfitting problem since the operations of generating hash codes for training data and hash function for test data are independently handled, which will lead to poor generalization ability. Minimal loss hashing (MLH) [7] that have shown higher search accuracy than unsupervised hashing approaches, but they all impose difficult optimization and slow training mechanisms. Spectral hashing (SpH) [13] uses a separable Laplacian eigenfunction formulation that ends up assigning more bits to directions along which the data has a greater range. However, this approach is somewhat heuristic and relies on an unrealistic assumption that the data is uniformly distributed in a high-dimensional rectangle.

With no regard of feature extraction, many research works have shown that the deep learning based approaches [5, 10] can provide a good performance in many image related tasks. Those methods normally need to use the autoencoders to pre-train a multiple-layer neural network, and then the pre-trained network in deep learning can be regarded as a representation function for new images, and the output vectors of those neural nodes can be viewed as a representation of the new images. Some deep learning method are proposed

*Hefei Ling is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '14 November 03 - 07 2014, Orlando, FL, USA
Copyright 2014 ACM 978-1-4503-3063-3/14/11
http://dx.doi.org/10.1145/2647868.2654987.

969
to implement image [4] or text [2, 8] retrieval task, these techniques push the retrieval field significantly forward.

Motivated by recent progresses in deep neural network and manifold learning, we will focus on the image representation learning, which is composed of a novel autoencoders based learning and a transfer learning based generation for image hashing and representation. We first construct an objective function for autoencoders to learn the hash function for training images, which not only consider the reconstruction error during encoding but also consider the neighborhood-structure of data. Then we minimize corresponding objective function in every layer of our multi-layer autoencoders. After that, we apply transfer learning to learn the hash code of new images for image representation. The rest of this paper is organized as follows: In section 2, we illustrate our new auto-encoder for image representation learning. In section 3, we detail the transfer learning based generation. Extensive experimental results are presented in Section 4. Finally, we provide a conclusion in Section 5.

2. NEIGHBORHOOD-STRUCTURE PRESERVED AUTOENCODERS

To discover the latent structure of the original data, the neighborhood-structure of the data should be preserved to guarantee its discrimination, which means that perceptually similar image should be mapped into similar hash code in hamming space and vice versa. Given training image data $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{p \times n}$, we want to get hash codes denoted by $H = [h_1, h_2, \ldots, h_n]^T \in \{0,1\}^{n \times c}$ ($c$ is the length of the hash codes). To exploit the local neighborhood structure of a data, we construct a n x n similarity matrix S as

$$S_{ij} = \begin{cases} 1 & \text{if } x_i \in N_k(x_j) \text{ or } x_j \in N_k(x_i) \\ 0 & \text{otherwise} \end{cases}$$

where $N_k(x)$ represents the set of k-nearest-neighbors of feature vector x.

The Hamming distance between two binary codes $h_i$ and $h_j$ (corresponding to data $x_i$ and $x_j$) is given by the number of bits that are different between them. To preserve the neighborhood structure, we seek to minimize the weighted average Hamming distance which represents the neighborhood-structure

$$\sum_{i=1}^{n} \sum_{j=1}^{n} S_{ij} \|h_i - h_j\|^2 = 2\text{tr}(H^T(N - S)H)$$

where $N_{ii} = \sum_{j=1}^{n} S_{ij}$ and other elements are zero.

Combine the neighborhood-structure preservation and reconstruction error, we get an unified objective function for our auto-encoder

$$\arg \min Q = \arg \min \|X - Q\|^2 + \lambda \text{tr}(H^T(N - S)H)$$

where $\lambda$ is a coefficient and

$$H = \text{sigmoid}(W_H X + b_H)$$
$$Q = \text{sigmoid}(W_Q H + b_Q)$$

By applying Broyden-Fletcher-Goldfarb-Shanno (BFGS) training algorithm, we get the learning parameter $W_H$ and $b_H$ for further hashing.

3. TRANSFERRING IMAGE REPRESENTATION

We present an approach that begins to learn a slight feature representation by using the unlabeled data $X_u$. By applying this learned representation to the labeled data $X_l$, we obtain a higher level representation of the labeled data, and an easier supervised learning task. To adapt our modified version of the sparse autoencoders algorithm to the new task and the new domain, we continue training the parameters by using a labeled dataset. Inductive transfer learning method is employed, different weight are assigned to the loss function of the target task to make sure that better performance in target domain can be achieved. The proposed method assumed that the parameters set $W$ and $b$ in an l-layer neural networks for target task can be divided into two parts. One is a share parameters set derived from the source task and the other is a special parameters set added in the target task.

$$W_T = \begin{cases} W_S & , 2 < l < k - 1 \\ W_{T'}, & k < l < n_i \end{cases}$$

$$b_T = \begin{cases} b_S & , 2 < l < k - 1 \\ b_{T'}, & k < l < n_i \end{cases}$$

where $W_S$ and $W_T$ are parameters set of the neural networks for the source task and the target learning task, respectively. $W_{T'}$ is specific parameters set for the target task. By assuming $h_{WS}(x) = f(W^T x + b)$ to be the activation function for source task $S$ and target task $T$, an extension of sparse autoencoders to target transfer learning case can be written as the following:

$$J(W, b) = \frac{1}{m} \left( \sum_{i=1}^{m} \frac{1}{2} \| h_{WS}(x^{(i)}) - y^{(i)} \|^2 \right) + \frac{\rho}{2} \left( \sum_{l=1}^{k-1} (W_S^{(l)})^2 + \sum_{l=k}^{n_1} (W_T^{(l)})^2 \right) + \beta \sum_{j=1}^{s} KL(\rho_j)$$

$$s.t. W = W_T, b = b_T, W_S^{(l)} \in \{S, T\}, W_T^{(l)} \in \{T\}$$

The first term in the definition of $J(W, b)$ is an average sum-of-squares error term, the second term is a regularization term that tends to decrease the magnitude of the weights, and helps prevent overfitting, the third term is sparsity penalty term. $\psi$ is the weight decay parameter which controls the relative importance of the weights, $\rho$ is a sparsity parameter (typically a small value close to zero), $\rho_j$ is the average activation of hidden unit $j$, $s$ is the number of neurons in the hidden layer, the index $j$ is summing over the hidden units in neural network. $\beta$ controls the weight of the sparsity penalty term. $W_S$ and $b$ are unchanged from the source task, by solving the optimization problem above, we can learn the parameter $W_T$ and $b_T$ simultaneously.

In order to achieve the transfer, in the target network, we remove the output layer of the source network and add several adaptation layers formed by fully connected layers that use the $W_T$ and $b_T$ as network parameters. Now, we can compute the parameters as follows:

970
1. Perform a feedforward pass, computing the activations for each layers respectively, using the equations defining the forward propagation steps.

2. For the output layer $n_l$, set: $\delta^{(n_l)} = -(y - a^{(n_l)}) \cdot f'(z^{(n_l)})$.

3. For layer $l = n_l - 1, n_l - 2, \ldots, k$, set:
   \[
   \delta^{(l)} = (W^{(l)} f^{(l+1)}) \cdot f'(z^{(n_l)})
   \]  
   (8)

4. For layer $l = k - 1, k - 2, \ldots, 2$, keep $W^{(l)}$ of target task equals the $W^{(l)}$ of source task, and set:
   \[
   \delta^{(l)} = ((W^{(l)}) f^{(l+1)}) \cdot f'(z^{(n_l)})
   \]  
   (9)

5. For layer $l = k - 1, k - 2, \ldots, 2$, keep $W^{(l)}$ and $b^{(l)}$ not change. For layer $l = n_l - 1, n_l - 2, \ldots, k$, implement parameter iteration of batch gradient descent, set:
   \[
   \Delta W^{(l)} := W^{(l)} - \delta^{(l+1)}(a^{(l)})^T
   \]  
   (10)

   \[
   \Delta b^{(l)} := b^{(l)} + \delta^{(l+1)}
   \]  
   (11)

By update the parameters, we can repeatedly take steps of gradient descent to reduce our target cost function $J(W, b)$ to train our neural network.

4. EXPERIMENTS

To validate that our approach Inductive Transfer Deep Hashing (ITDH) achieves better retrieving performance, we compare it with Locality Sensitive Hashing (LSH)[1], Semi-Supervised Hashing (SSH)[12], Spectral Hashing (SH)[13] and Restricted Boltzmann Machines (RBMs)[3] on the MNIST digit dataset and STL-10 dataset in this section. We first using the transfer deep learning algorithm to extract image feature, then transform it into hash codes, and through the hamming-distance between the query sample and all training samples to measure the similarity; then according to the similarity sorting the query results, finally, we using precision-recall curve to measure the performance of the algorithms.

4.1 MNIST Digits

The MNIST dataset consists of a total of 70000 handwritten digit sample, each of size 28×28 pixels. Each sample is associated with a label from 0 to 9. Our goal is to distinguish between the digits from 0 to 4. We use the digits 5 to 9 as our “unlabeled” dataset with which to train the source neural network, then use a labeled dataset which consists 15298 digits from 0 to 4 to train the target neural network. The rest of the remaining 15298 digits from 0 to 4 are used to test the efficiency of our algorithm. RBMs train neural networks using the unlabeled data for training and using the labeled pairs for fine-tuning. Our network is designed to be very simple, the source architecture consists of two stack autoencoders layers, one full-connection layer and one output layer(for extract hash codes), such as 200-200-400F-output. The target architecture similar to source architecture, just added a full-connection before the output layer. Given a dataset, we use each sample in the test set as a query to retrieve samples in the training set within a specific Hamming distance(which is fixed at 0 in our experiment), and then compute standard retrieval performance measure: Precision-Recall curve and retrieving accuracy. To determine whether a retrieved object is relevant to the given query, we adopted the evaluation methodology that retrieving same category samples which are considered as ground-truth relevant samples.

Figure 1a shows the retrieving accuracy of the proposed approach(ITDH) compared to SSH, SH, LSH and RBMs algorithm. In order to show the influences of code length on the retrieving performance, we vary the code length from 8 to 256 bits. We first find all the relevant objects whose Hamming distance to the query equals 0, then consider the most frequently appearing label responding to the relevant objects as the label of the query. Finally accuracy is computed by the ratio that the retrieval label equals the original one. Compared with other algorithms, the retrieval accuracy of our approach is 1.5 times as other methods. In addition, the Precision-Recall curve for 48-bit codes is shown in Figure 1b, which clearly demonstrates proposed approach achieve the best performance over the entire Hamming ranking space.

Figure 1c shows the Precision-Recall curve of difference code Length. With the increase of the code length, the image retrieval performance increases. It is chiefly because that the more hidden layer neurons can learn the rich feature which improve the ability of image recognition in deep network. Since the number of the neurons of the last hidden layer is the code length of image, the length of code directly impact on the final image retrieval performance. On the MNIST dataset, when increasing the number of the neurons of the last hidden layer from 8-bit codes to 48-bit codes, algorithm performance is tending towards stability.

We provides the training and testing time (including feature learning and retrieval) for different methods. Clearly, RBMs and ITDH is most expensive to train, RBMs needs at least 3.5 orders of magnitude more time than other algo-
rithms, while ITDH needs at least 2.5 orders of magnitude more time. LSH needs negligible training time since the projections are randomly generated, instead of being learned. SSH and SH take need a bit more time to training than LSH. Summarize, the code generation time can be ranked as: RBMs $\gg$ ITDH $\gg$ SH $\approx$ SSH $>$ LSH. In terms of test time, RBMs is still most expensive because it computes the binary codes for a query using multi-layer neural network. ITDH is a neural network, but due to the simple structure, it take similar test time of others. LSH are the fastest techniques. SH and SSH require a little more time than LSH due to the computation of nonlinear sinusoidal functions to generate each bit. So, the test time can be ranked as: RBMs $\gg$ ITDH $\approx$ SH $\approx$ SSH $>$ LSH.

4.2 STL-10 dataset

The STL-10 dataset is an image recognition dataset for developing unsupervised learning algorithms. It has 10 classes, 5000 training images (500 images per class), 8000 test images (800 images per class) and 100000 unlabeled images. We use the unlabeled images set for source task to learn image features and apply the representations for target task to train train set and test set. Figure 2 shows the quantitative evaluation of different methods. The proposed approach outperform the other methods in most of the cases performing significantly better than all the compared methods for higher bits as expected.

5. CONCLUSIONS

The main contribution of this paper is a novel Inductive Transfer Deep Hashing (ITDH) approach for semantic hashing based fast similarity search. The proposed method combines the neighborhood-structure of data with a transfer learning algorithm to learn the hash function for training images, which is formulated as minimizing empirical error while maximizing variance and independence of hash bits over the labeled and unlabeled data. By solving the problem of finding small codes for large data into two stages: unsupervised learning and supervised learning, we achieve a flexibility algorithm architecture. Due to neural net can be viewed as a way for learning feature expression, each neuron can be regarded as a cell of image sample. Through the learning by a large number of unlabeled data, can make the cells of images more ordinary, while by learning labeled data, neural net can better combine with these cells to express the target image. The experiments on two image datasets show that the proposed approach outperforms several existing state-of-the-art techniques. In the future, we shall apply this technique to other complex benchmark dataset and achieve even higher effectiveness and efficiency by employ more powerful unsupervised or supervised learning algorithms. It would also be interesting to combine semantic hashing with GPU computing(or distributed computing) to further improve the speed and scalability of similarity search.

6. ACKNOWLEDGMENTS

This work is supported by the NSF of China under Grant No.61272409, the applied basic research of youth-project of Yunnan Provincial Science and Technology Department under Grant No.2012FD0034.

7. REFERENCES