Nearest Neighbor Lookup: Iterative quantization of feature vector from the final layer and do a nearest neighbor lookup

Group Members
K Mahidhar
Rajith Jarang
Boddu Prasad

Abstract

In this we use last layer of neural network to learn similarity-preserving binary codes for efficient similarity search in large-scale image collections.To reduce the quantization error we use a method iterative quantization(ITQ) in which we find a orthogonal matrix for applying rotation to the zero centered data.

1 Introduction

We learn similarity preserving binary codes for large scale image properties. An effective implementation must have the following properties:
The codes should be short to enable the storage of large amounts of images in memory. and than the codes map images that are similar (either in terms of feature space distance or semantic distance) to binary strings with a low Hamming distance. Learning the parameters of the binary code and for encoding a new test image should be efficient and scalable.

2 Methods

A common initial step in many binary coding methods is to perform principal component analysis (PCA) to ensure good generalization ability for small code sizes. Since the variance of the data in each PCA direction is different in general, encoding each direction with the same number of bits is bound to produce poor performance. There are several methods that deal with this like Spectral Hashing which uses a separable Laplacian eigenfunction formulation that ends up assigning more bits to directions along which the data has a greater range.Semi-supervised hashing (SSH) relaxes the orthogonality constraints of PCA to allow successive projection directions to capture more of the data variance.
Applying a random orthogonal transformation,does a very good job of balancing the variance of different PCA directions and outperforming both Spectral hashing and non-orthogonal relaxation(SSH). For updating the initial orthogonal transformation to reduce the quantization error iteratively we use Iterative quantization approach(ITQ).

3 Iterative quantization

Let the matrix after applying pca be V and B be the binary code matrix. Applying a random orthogonal transformation dosen't change the optimal solution and We begin with a random initialization of the orthogonal matrix(R). In each iteration, each data point is first assigned to the nearest vertex(B) of

Submitted to 29th Conference on Neural Information Processing Systems (NIPS 2016). Do not distribute.

the binary hypercube, and then R is updated to minimize the quantization loss given this assignment. We should minimize the quantization loss $Q(B, R) = \|B - VR\|_F^2$, where $\|.\|_F$ denotes frobenuis norm. - (1)

3.1 Fix rotational matrix(R) and than update B

Expanding (1) we get,$Q(B, R) = \|B\|_F^2 + \|V\|_F^2 - 2tr(BR^T V^T)$ - (2)
Minimizing the above equation is same as maximizing
$tr(BR^T V^T) = \sum_{i=1}^{n} \sum_{j=1}^{c} B_{ij}\tilde{V}_{ij}$,where,$\tilde{V}_{ij}$ is the element of the matrix VR - (3).
To maximize the above equation,$B_{ij} = 1$ whenever $\tilde{V}_{ij} \geq 0$ and -1 otherwise.Here,$\{-1, 1\}$ are used intead of $\{0, 1\}$ to center data which are later converted to $\{0, 1\}$.This is same as applying signum function to VR i.e B = sgn(VR)
Using the above formulation we update B.

3.2 Fix B and than update rotational matrix(R)

Expanding (1) we get,$Q(B, R) = \|B\|_F^2 + \|V\|_F^2 - 2tr(BR^T V^T)$ - (2)
In this to minimize the quantization error a rotation is found which align one point with another. The SVD of c*c matrix $B^T V$ is used which is $S\Omega\hat{S}^T$ and than we assign the rotational matrix R to $\hat{S}S^T$ i.e $R = \hat{S}S^T$
Using the above formulation we update R.

4 Extracting feature vector

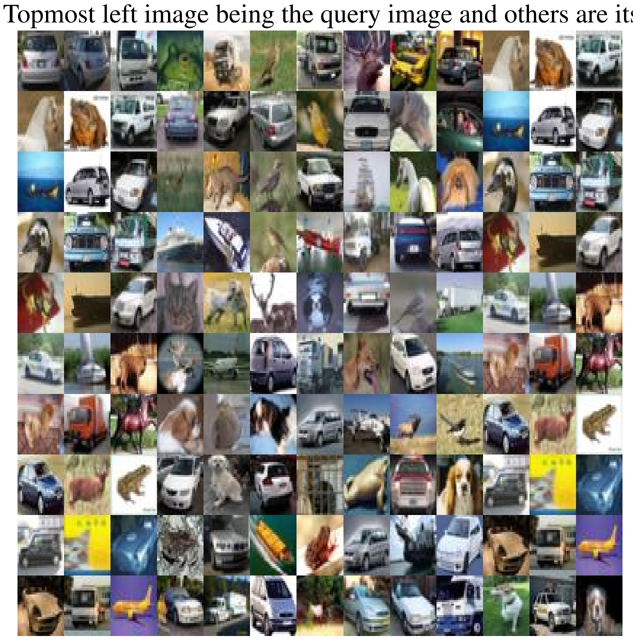
We use inception model which has been trained on ILSVRC-2012 images, which is a different dataset than CIFAR-10 to extract features. The network to has been trained for the 1000 classes of the ILSVRC-2012 dataset but instead of taking the last layer - the prediction layer - we use the penultimate layer.This layer can be regarded as a representation layer, which represents images as 2048 dimensional feature vectors. We also used gist descriptor to test the method.

5 Nearest Neighbour lookup

We take the top 130 neighbours for each in the test data of the CIFAR 10 dataset to calculate the precision for different code sizes and printing them.

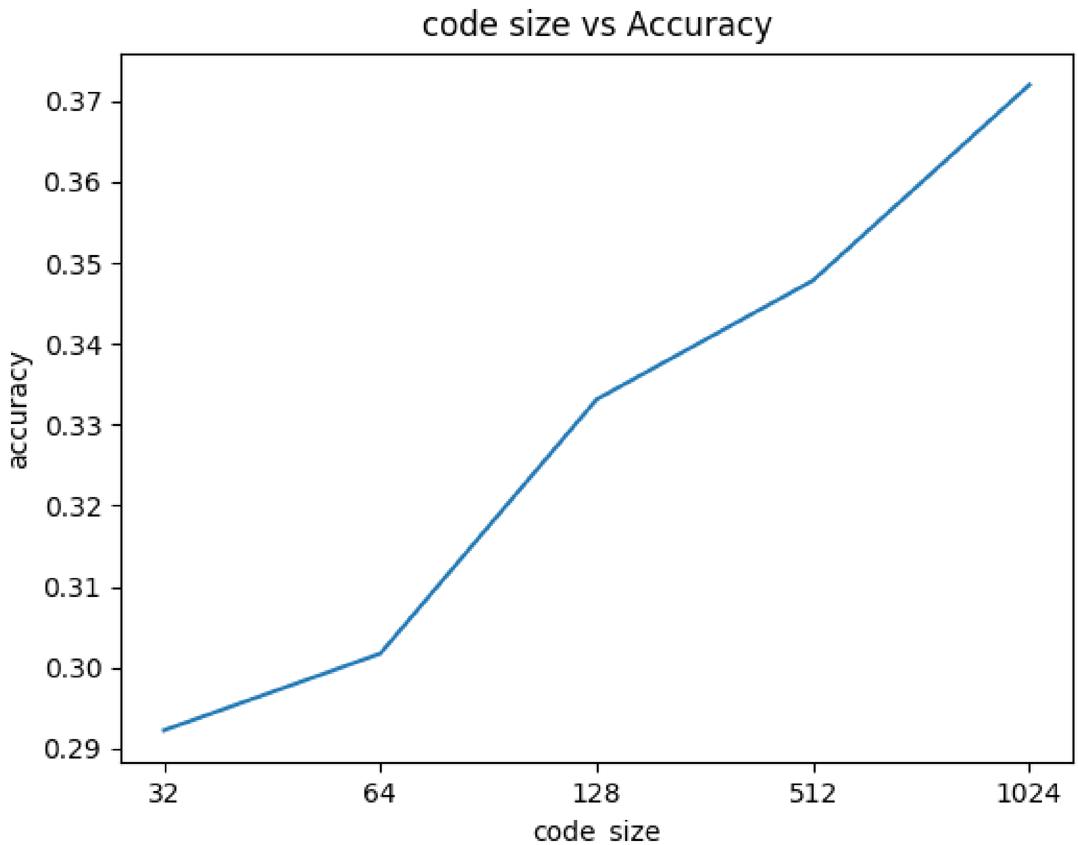
5.1 Output image

output below is after taking the second element in the test data and printing its top 130 nearest neighbours.The Topmost left image being the query image and others are its neighbours.



5.2 Accuracy

We calculated the classification accuracy across the test data for the top 130 nearest neighbours.The graph is as follows:



6 Inferences

We observe that although the code size is 32 which is reduced from 2048 vector has close accuracy to that of code size of 1024.From this we can observe that this method works better for low code sizes.We also observe that the feature vector obtained from the neural network give a better accuracy than gist descriptor for the same code size.

References

[1]Yunchao thoGong, Svetlana Lazebnik, Albert Gordo, Florent Perronnin. Iterative Quantization : A Procrustean Approach to Learning Binary Codes for Large-scale Image Retrieval