DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DEEP SUPERVISED HASHING FOR FAST IMAGE RETRIEVAL

April 15, 2018

Aarathi M (20173018)
Rohit Tiwari (20162085)

Introduction

Number of images that are available online are increasing at an exponential rate every year. In order to build a robust image search application to find images relevant to the query image, it is required to match the image with the huge image dataset that is available. Convolution Neural Networks (CNN) is currently being used in image classification, object detection and analyzing visual imagery. To solve this problem of faster image search, a CNN based hashing method can be used. This project implements this method based on the CVPR paper âĂIJDeep Supervised Hashing for Fast Image RetrievalâĂİ. Using this method, each image will be converted to k-bit binary code to preserve the information of similarity.

Approach

CNN based approach to learn discriminative image representations and compact binary codes simultaneously is used for this purpose. This method first trains the CNN using image pairs and the corresponding similarity labels. The loss function is designed to learn similarity-preserving binary codes.

Implementation

The objective of this project is to build a CNN based hashing method to convert image to a binary number for faster search. Keras framework is used with tensorflow as backend for this purpose.

Architecture:

The model from image to k-bit binary outputs is called as Model-1. Model-1 contains the following layers. The architecture diagram is shown in Figure 2.

1. Conv (5x5x32), strides =1, Activation = Relu
2. Max pooling 3x3x32, strides =2
3. Conv (5x5x32), strides =1, Activation = Relu
4. Average pooling 3x3x32, strides = 2
5. Conv (5x5x64), strides =1, Activation = Relu
6. Average pooling 3x3x32, strides =2

7. Fully connected layer with 500 nodes, activation = Relu

8. Fully connected layer with 12 nodes, activation = sigmoid

The above model Model-1 gives 12 outputs which can be converted to binary by applying a threshold of 0.5.

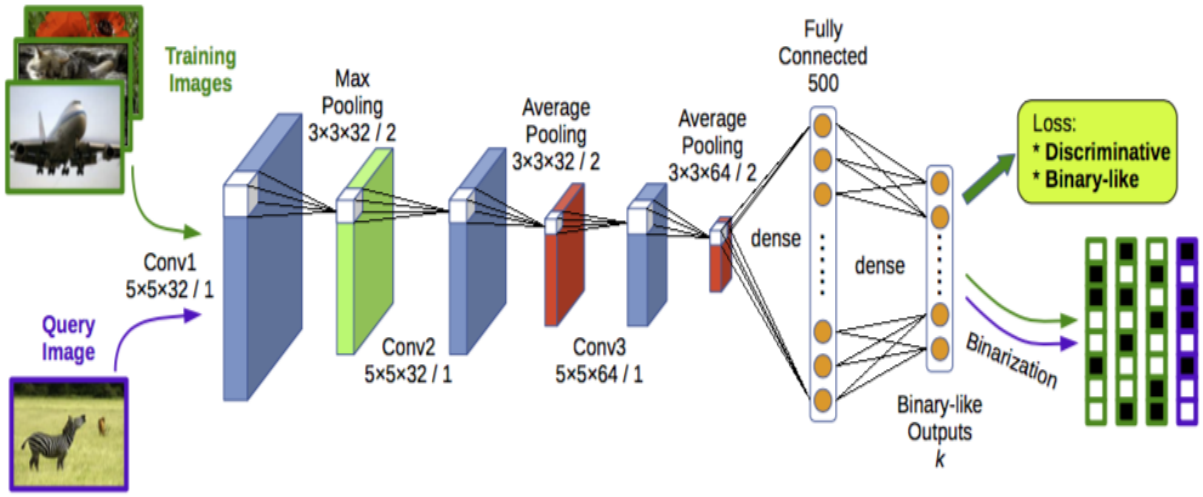

Figure 1: Network Structure

Dataset

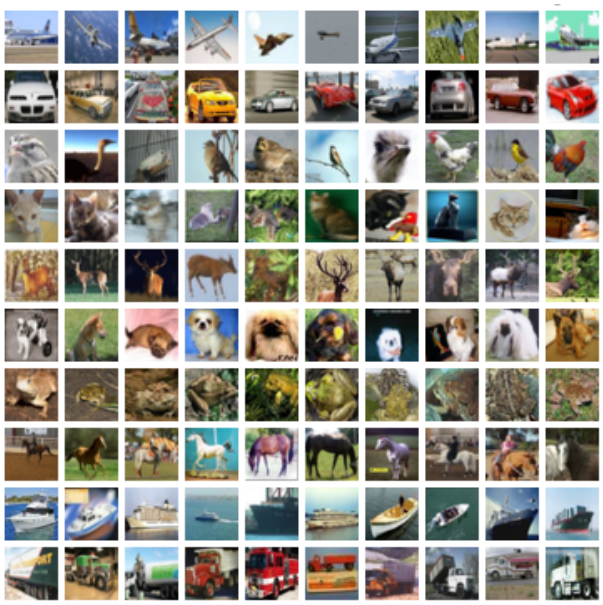

Figure 2: CIFAR-10 Dataset

CIFAR-10 is the dataset that is chosen for this project. CIFAR-10 contains 50,000 images belonging to 10 categories each of size 3*32*32. Above figure shows some of the sample images from the dataset.

Training Methodology:

As discussed above, the training dataset contains 50,000 images belonging to 10 categories. As our problem is to identify similar and dissimilar images, all the images that belong same class are identified as similar and images that belong to different classes are identified as dissimilar images. To achieve this, image pairs are selected randomly from the dataset and depending on the image labels, the image pair are labeled similar or dissimilar. The idea is to keep the difference between Model-1 binary outputs smaller for similar images and farther for dissimilar images. The outputs obtained from Model-1 are merged and the output is used to measure loss. For merge function, we built two models one with L1 norm and the other with L2 norm with and without regularization. The results and are shown below

Loss Function defined in the paper

The loss function such that the codes of similar images should be as close as possible, while the codes of dissimilar images being far away. Based on this objective, the loss function is naturally designed to pull the codes of similar images together, and push the codes of dissimilar images away from each other. The loss with respect to the pair of images is defined as:

$$\begin{aligned} L(b_1, b_2, y) = &\frac{1}{2}(1-y)D_h(b_1, b_2) \\ &+\frac{1}{2}y\ \max(m - D_h(b_1, b_2), 0) \\ &s.t. \quad b_j \in \{+1, -1\}^k, j \in \{1, 2\} \end{aligned}$$

where $D_h(.,.)$ denotes the Hamming distance between two binary vectors, and m > 0 is a margin threshold parameter.
Suppose that there are N training pairs randomly select- ed from the training images,our goal is to minimize the overall loss function:

$$\begin{aligned} \mathscr{L} = &\sum_{i=1}^{N} L(b_{i,1}, b_{i,2}, y_i) \\ &s.t. \quad b_{i,j} \in \{+1, -1\}^k, i \in \{1, ..., N\}, j \in \{1, 2\} \end{aligned}$$

Relaxation

A regularizer is imposed on the real-valued network outputs to approach the desired discrete values (+1/-1). To be specific, we replace the Hamming distance in Eqn.(1) by Euclidean distance, and impose an additional regularizer to replace the binary constraints, then Eqn.(1) is rewritten as:

$$\begin{aligned} L_r(b_1, b_2, y) = & \frac{1}{2}(1-y)||(b_1 - b_2||_2^2 \\ & + \frac{1}{2} y \ \max(m - ||(b_1 - b_2||_2^2, 0) \\ & + \alpha(||\ |b_1|-1||_1 + ||\ |b_2|-1||_1) \end{aligned}$$

By substituting Eqn.(3) into Eqn.(2), we rewrite the relaxed overall loss function as follows:

$$\begin{aligned} \mathscr{L}_r = \sum_{i=1}^{N} \{ & \frac{1}{2}(1-y_i)||(b_{i,1} - b_{i,2}||_2^2 \\ & + \frac{1}{2} y_i \ \max(m - ||(b_{i,1} - b_{i,2}||_2^2, 0) \\ & + \alpha(||\ |b_{i,1}|-1||_1 + ||\ |b_{i,2}|-1||_1)\} \end{aligned}$$

Results

case-1

- Number of neurons in output layer : 12
- The outputs of Model-1 for two images (b1-b2) are merged by performing element-wise subtraction and taking the absolute value of the difference.
- Fully connected layer with one neuron is added to this layer.
- This model is trained with the training dataset created earlier.
- Number of Image pairs: 200000
- Number of epochs: 30
- Batch Size: 200

Where b1,b2 are binary outputs of two images in an image pair and y is the similarity measurement. Y value is 0 for similar images and 1 for dissimilar images. Image pairs are

chosen randomly from the CIFAR-10 image set of 10 classes. Because of this the number of dissimilar images chosen will be approx 9 times more than similar images. To solve this problem, 70 percent of the dissimilar images are removed from the image pair selection.

Following are the accuracy and loss curves for training and validation dataset for epochs 1 to 100.

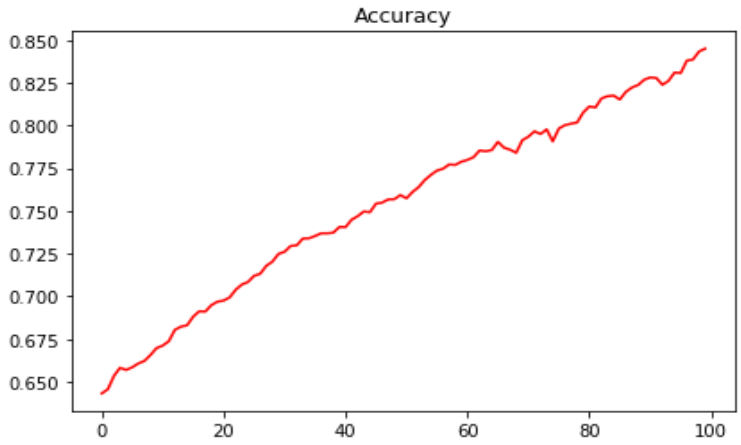

Figure 3: Accuracy curve for training and validation dataset

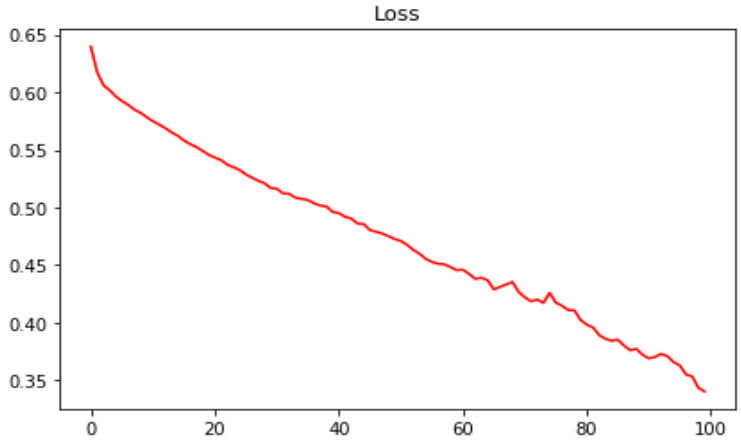

Figure 4: Loss curve for training and validation dataset

Following are the binary values generated for some of the images from the test dataset.

- 100000000001 Frog
- 010011110000 Truck
- 011011110010 Truck
- 100000000001 deer

- 010001110000 automobile
- 111011000000 automobile
- 100100001101 bird
- 100100000100 horse
- 111010010010 ship
- 100000000000 cat
- 100000000001 deer

Precision = 0.7176 Recall = 0.7268 Average precision = 0.6942

case-2

- Number of neurons in output layer : 12
- The outputs of Model-1 for two images (b1-b2) are merged using euclidean distance.
- Loss function discussed under the loss function is implemented
- Margin (m) =24 and L1 Regularizer (alpha) = 0.01
- Number of Image pairs: 200000
- Number of epochs: 30
- Batch Size: 200

Where b1,b2 are binary outputs of two images in an image pair and y is the similarity measurement. Y value is 0 for similar images and 1 for dissimilar images. Image pairs are chosen randomly from the CIFAR-10 image set of 10 classes. Because of this the number of dissimilar images chosen will be approx 9 times more than similar images. To solve this problem, 70 percent of the dissimilar images are removed from the image pair selection.

Following are the loss curves for training and validation dataset for epochs 1 to 30.

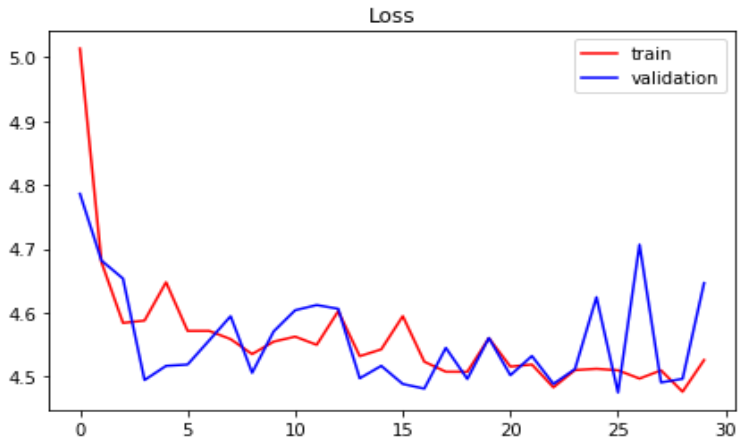

Figure 5: Loss curve for training and validation dataset

Loss function discussed in the paper is implemented in this case. From the above figure, it can be observed that the loss values initially decreased till 4 epochs and remained more or less constant. This needs to be improved further.