Generalized Product Quantization Network for Semi-supervised Image Retrieval
- Supplementary Material -

Figure 1. The results of hyper-parameter variants.

<table>
<thead>
<tr>
<th>Value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha)</td>
<td>1 5 20 80</td>
</tr>
<tr>
<td>(\beta)</td>
<td>1 2 4 8</td>
</tr>
<tr>
<td>(K)</td>
<td>2^2 2^4 2^6 2^8</td>
</tr>
<tr>
<td>(R)</td>
<td>1/8 1/4 1/2 1/1</td>
</tr>
</tbody>
</table>

Table 1. A list of Hyper-parameter values.

Analysis on Hyper-parameters As mentioned in section 4.2 of our original submission, we present additional information regarding hyper-parameters: \(\alpha, \beta,\) and \(K\). Besides, to see how GPQ actually uses unlabeled data, we set up a hyper-parameter \(R\) that determines the ratio of the total unlabeled data used for training. The default value is set as \(\{\alpha, \beta, K, R\} = \{20, 4, 2^4, 1/1\}\) in order, and we vary each hyper-parameter as listed in Table 1 while fixing others at defaults. We conduct experiments on CIFAR-10 and NUS-WIDE datasets for the binary code of 48-bits.

From Figure 1, we can see that trends of mAP scores according to variation of \(\alpha\) and \(K\) are similar to those observed in other deep quantization methods [1, 2]. The hyper-parameter \(\beta\) that controls the randomness of predictions before applying softmax shows optimal performance at 4. The results of \(R\) related with the amount of unlabeled data justify that GPQ can fully utilize the unlabeled data to improve retrieval performance.

Algorithm We demonstrate our training process in Algorithm, where \(\gamma\) denotes learning rate. The result of entropy maximization and minimization can be observed in updating stage of \(\theta_C\) and \(\theta_F\), respectively. Gradients generated from the unlabeled data in a batch do not flow directly into \(\theta_Z\), hence, previous ones can be reflected by initializing \(\theta_Z\) from \(\theta_C\).

Algorithm 1 GPQ training for batch size \(B\)

Input: Parameters of each component: \(\theta_F, \theta_Z, \theta_C\)

Input: Batch \(B = \{(I^L_1, y_1, I^U_1), ..., (I^L_B, y_B, I^U_B)\}\)

1. Initialize \(\theta_Z\) with \(\theta_C\) by soft assignment
2. for \(i = 1 \ldots B\) do
3. \(\hat{x}_i^L, \hat{x}_i^U \leftarrow F_{\theta_F}(I^L_i, I^U_i)\)
4. \(\hat{q}_i^L \leftarrow Z_{\theta_Z}(\hat{x}_i^L)\)
5. \(\hat{p}_{i}^L, \hat{p}_{i}^U \leftarrow C_{\theta_C}(\hat{x}_i^L, \hat{x}_i^U)\)
6. end for
7. if label=true then
8. \(\ell_{N.PQ} \leftarrow \mathcal{L}_{N.PQ}\) with \(\{\hat{x}_i^L, \hat{q}_i^L, y_i\}_{i=1}^B\)
9. \(\ell_{cls} \leftarrow \mathcal{L}_{cls}\) with \(\{\hat{p}_i^L, y_i\}_{i=1}^B\)
10. else
11. \(\ell_{SEM} \leftarrow \mathcal{L}_{SEM}\) with \(\{\hat{p}_i^U\}_{i=1}^B\)
12. end if
13. \(\theta_F \leftarrow \theta_F - \gamma \left(\frac{\partial \ell_{N.PQ}}{\partial \theta_F} + \frac{\partial \ell_{cls}}{\partial \theta_F} + \frac{\partial \ell_{SEM}}{\partial \theta_F}\right)\)
14. \(\theta_Z \leftarrow \theta_Z - \gamma \frac{\partial \ell_{N.PQ}}{\partial \theta_Z}\)
15. \(\theta_C \leftarrow \theta_C - \gamma \left(\frac{\partial \ell_{cls}}{\partial \theta_C} + \frac{\partial \ell_{SEM}}{\partial \theta_C}\right)\)

Output: Updated \(\theta_F, \theta_Z, \theta_C\)

References