Deep Learning in Practice - Ecole CentraleSupelec
Spring 2019
Assignment 2 (Naver Labs)

Ayush K. Rai, Kai-Wei TSOU
ayush.rai2512@student-cs.fr,kai-wei.tsou@supelec.fr

March 3, 2019

Solution : Question 1

What does each row of the matrix feats represent?

The rows of the matrix feats represent the fc7 features (output of fc7 layer in the AlexNet Architecture) for every image in the dataset.

Solution : Question 2

Where does the dimension of these lines comes from and how do we extract these features?

The AlexNet architecture is designed in such a way that the output of fc7 (fully connected layer or Linear Layer) is a 4096 length vector.

Solution : Question 3

What can be observe from the t-SNE visualization? Which classes 'cluster' well? Which do not?

The radcliffe_camera and chirst_church cluster quite well but the other classes have overlap in the 2-dimension space. For example, the class keble and cornmarket seem to be diverse and have no clear cluster.

Solution : Question 4

Should we get better results? What should change? Why?

If we finetune the AlexNet model trained on ImageNet dataset (1000 classes) on the Landmark dataset (586 classes) we have two arguments.

- Since CNNs have the property of Compositionality, therefore the weights in the lower convolutional layers, which usually capture edge related information will be initialized in a better way with finetuning AlexNet (trained on ImageNet) on Landmark dataset than training AlexNet on Landmark dataset from scratch.
- Another important aspect to consider is that ImageNet dataset has images which are very different to Oxford Dataset, whereas images in the Landmark dataset are more similar to images in the Oxford building dataset.

For these important reason, finetuning is crucial and will produce better results.

Solution : Question 5

Why do we change the last layer of the AlexNet architecture?

Since we have only 586 classes in Landmark dataset while in the ImageNet there are 1000 classes, so we should change the number of neurons in the last fc layer in order to train the AlexNet.

Solution : Question 6

How do we initialize the layers of model_1b for finetuning?

For early layers (dealing with local features) of model_1b we can keep the same weights in model_1a and for deeper layers, we can perform xavier initialization.

Solution : Question 6

How does the visualization change after finetuning? What about the top results?

We can observe the classes cluster slightly better. For example, the clusters of magdalen and clase bodleian can be clearly observed without much overlapping with other classes. However, the classes such as all_souls and hertford still overlap with each other. For the top-15 results, we found the average precision is higher than the AlexNet trained with ImageNet.

Solution : Question 7

Why images need to be resized to 224x224 before they can be fed to AlexNet? How can this affect results?

Since the AlexNet Architecture has predefined sizes for convolutional layers (padding size, strides, kernel size), pooling layers and fully connected layers, therefore we need to resize images into the specified dimension of 224×224 (to match dimension required by the architecture) before using them as input to the neural architecture. The disadvantage of resizing images is that we might loose essential information and image quality due to it.

Solution : Question 8

Why does the size of the feature representation changes?

The generalized-mean (GeM) pooling is a global pooling method which performs the operation as follows:

$$\mathbf{f}^{(g)} = [f_1^{(g)}, \ldots, f_K^{(g)}], \;\; f_k^{(g)} = \left(\frac{1}{|\mathcal{X}_k|} \sum_{x \in \mathcal{X}_k} x^{p_k} \right)^{\frac{1}{p_k}}$$

where $\mathcal{X}_k$ is the two-dimensional features in channel k. Note that the pooling is performed over global features for each channel and results in one scalar value for each channel. In our case, we have the features of shape $N \times N \times 256$ before GeM, so the size of resulting features would be 256.

Solution : Question 9

Why does the size of the feature representation is important for a image retrieval task?

Image retrieval is simply performed by exhaustive euclidean search over database descriptors w.r.t. the query descriptor. This is equivalent to the inner product evaluation of l_2 normalized vectors, i.e. vector-to-matrix multiplication, and sorting. This operation is computationally expensive and therefore size of the feature representation is highly important.

Apart from the issue of computational time, another crucial point to consider is that conventional pooling strategies like max and mean pool were used as building block for neural networks for Image Classification tasks. These pooling strategies capture local features of each activation map and introduce scale invariance.

Whereas in Image Retrieval task, having a compact representation of the entire image is more important. GeM pooling strategy handles this issue by performing pooling over global feature of each activation map.

For example, in the tutorial by using conventional pooling techniques we had a feature representation of 4096 for every image whereas with GeM pooling layer the feature representation reduces to 256. This 256 dimension might be effective in faster and more accurate image retrieval.

The authors of GeM paper further endorse the idea of using GeM layer to learning representations (of sizes 256,512 and 2048) for images for Image Retrieval task outperforms conventional strategies.

Solution : Question 10

How does the aggregation layer changes the t-SNE visualization?

We can clearly observe that the clusters are more compact than previous examples. In the previous examples, the points with the same class scatter over large region while in this example, the points come together. We also observe that by using the aggregated layer the KL divergence reduces to 2.42 to previous 2.44 after 300 iterations.

Solution : Question 11

Can we see some structure in the clusters of similarly labeled images?

We observe the cluster for classes like **radcliffe_camera**, **pitt_rivers** and **hertford** have become more dense with few outliers. But we also observe that there are many overlapping clusters like **hertford** and **commarket** etc.

Solution : Question 12

Why do we change the average pooling layer of the original Resnet18 architecture for a generalized mean pooling?

First, after changing into GeM pooling, we are able to have more compact representations for each image. Second, the global max pooling and global average pooling are special cases of GeM pooling When $p_k \to \infty$ and $p_k = 1$ respectively. The parameters $\{p_k\}$ can be manually set or learned in the training phase. So GeM is more flexible since it could choose between max-pooling and average-pooling according to the training dataset.

Solution : Question 13

What operation is the layer model_1d.adpool doing?

The generalized-mean (GeM) pooling performs the operation as follows:

$$\mathbf{f}^{(g)} = [f_1^{(g)}, \ldots, f_K^{(g)}], \; f_k^{(g)} = \left(\frac{1}{|\mathcal{X}_k|} \sum_{x \in \mathcal{X}_k} x^{p_k} \right)^{\frac{1}{p_k}}$$

Where $\forall k, p_k = 3$.

Solution : Question 14

How does this model compare with model 1c, that was trained in the same dataset for the same task?

The average precision of the model 1c is better than that of the model 1d. The reason might be the ResNet18 overfits to the training data and becomes less general. So the precision is worse than that of model 1c. For the t-SNE, both models possess compact clusters thanks to lower dimension features. Specifically, we have slight lower KL divergence in the model 1d (2.35) the KL divergence of the model 1c (2.42).

Solution : Question 15

How does is compare to the finetuned models of 1b?

The average precision of the model 1d is better than that of the model 1b since the model 1d have more compact feature representations. We obtain lower KL divergence from the model 1d than the model 1b. Visually, the cluster of the model 1d is much more compact than that of the model 1b.

Solution : Question 16

What can we say about the separation of data when included unlabeled images?

The distribution of the unlabeled features scatter all over the 2-dimensional domain and have no specific cluster. We also observed that the KL divergence after including the unlabeled data has increased to 2.9.

Solution : Question 17

And the distribution of the unlabeled features?

The two features for unlabeled data points have strong negative correlation. However we do not observe any such correlation for labeled data points.

Solution : Question 18

How can we train a model to separate labeled from unlabeled data?

As we noticed in Question 17 (above) that unlabeled data points have strong negative correlation whereas labeled data points do not have any such relation. We can exploit this property and in addition we can also investigate some other discriminative features between labeled and unlabeled data and train a classifier such as SVMs or Random Forest to determine whether a point belongs to labeled data points set or unlabeled data points set. It would also be interesting to look into representation learning methods to calculate those discriminative methods on their own without requiring domain expertise.

Solution : Question 19

Compare the plots with unlabeled data of the model trained for retrieval (with triplet loss) and the model trained for classification of the previous subsection. How do they change?

The KL divergence decrease from 2.91 to 2.47, which means the data points with the same classes cluster closer. We can easily observe the the plots. For example, the class magdalen and cornmarket seem to have lower standard deviation.

Solution : Question 20

What is the difference in AP between a model that has trained with and without data augmentation?

We make the observation that Average Precision before data augmentation was 55.24 and after data augmentation it is 58.36.

Solution : Question 21

What about the clustering? Why do you believe some of the classes have not been adequately clustered yet?

From the 2 dimensional t-sne plot, we observe that certain images have been isolated from the rest of the cluster. We also notice that in some cases like **christ_church** there are two sub-clusters representing this class, which gives us a concrete evidence that classes have been adequately clustered.

Solution : Question 22

What other data augmentation or pooling techniques would you suggest to improve results? Why?

We can apply intensity transformations (log transformation, power law transformation) on images, add salt-pepper noise, blurring in the original images. These transformation will make the dataset more closer to real images and therefore make the learning model more robust and efficient.

Some advance work on Generative Adversarial Networks can be used to synthesize images which are a representative of the dataset but this is still an active area of research.

Solution : Question 24

Why using a larger architecture results in a higher AP? Is this always going to be the case?

We believe that the larger architecture of (ResNet-50) achieves a higher AP than that of smaller architecture (ResNet-18) because the ResNet-18 was underfitting the data and therefore there was a problem of high bias. ResNet-50 has more capacity because of more layers and also skip-connections, which improves gradient flow during training. Because of these reasons the ResNet-50 was able to reduce the bias of the ResNet-18 Model and get higher AP score.

However, we cannot generalize the statement that larger architecture will always achieve better AP than smaller architecture. Infact this has more to do with understanding bias-variance trade off. If the model is underfitting (high bias) the data then it makes sense to increase its capacity but if a model is already overfitting (high variance) the data then it makes more sense to increase the size of the training dataset.