STA 141C: Landmark Recognition Report

Huachao Lin Shuli Hu Miao Hu

Introduction

In an effort to better understand the principles and techniques of machine learning, our group decided to follow an on-going Kaggle competition and work on the task: Google Landmark Recognition 2019. In this competition, the participants are required to build models that recognize the correct landmark (if any).

Computer vision is one of the most prominent aspects of artificial intelligence and machine learning studies. Our motivation is to gain insights on this subject by having hands-on experience of extracting image features and attempting different classification techniques.

Our analysis indicates that the most robust model is a combination Mini Batch k-means with 500 clusters and 3-layer neural network with the number of epochs equal to 5. With this model, we are able to achieve a 53.3% accuracy rate.

Methodology

Our approach for image recognition consists of three major parts: feature extraction, representation simplification, and classification. The detailed descriptions of related algorithms are as follows.

- Feature extraction

  SIFT

The scale-invariant feature transform (SIFT) is a feature algorithm that extracts keypoints and computes descriptors from images. The algorithm has four steps: scale-space extrema detection, keypoint localization, orientation assignment, and keypoint descriptor. For each image input, SIFT will return 128-dimensional vectors as a descriptor.

- Representation simplification

  Bag-of-words

Bag of words model is a common natural language processing techniques. Here, we consider each image as a document of SIFT “words”. We can now extend the bag-of-words model to classify images instead of text documents. The Mini Batch K-means is adopted as the encoding method. A detailed algorithm is presented below:

- Classification

  Neural network

A neural network is a connection which one layer connect to another and the connections are weighted. It receives the input as neurons and trains the active function that can produce an output. We can use the neural network as a classification method for image retrieval. The basic idea is it can gather the unlabeled data together according to similar input.

Logistic regression

Logistic regression will transform the output by using the sigmoid function to the probability value that can map it to one of the classes we had. Since the sigmoid function is not linear, we need to change the cost function by using MSE to the Cross-entropy function which can easily be divided into two separate cost functions. The two separate functions are monotonic functions which can calculate the gradient and minimize cost. After we had minimized cost, we can find the optimal global minimum. So it can map the probability to the correct class.

- Dataset

The original dataset provided by the Kaggle website is a list of 672,036 images, each with a unique photo-id and one of the 140,839 landmark_ids. To accommodate the computation ability with the personal computer and the provided server, we filtered out the top 20 most frequently appeared landmarks from the original dataset. The dataset was reduced to 50,375 images, each with a unique photo-id and one of the 20 landmark-ids.

Implementation Details

1. Data preparation

We first got the feature matrix of the selected images by plugging the dataset in the SIFT algorithm under openCV. Next, the rows of the matrix that does not contain descriptors are dropped, resulting in a 1,654,284 x 128 feature matrix. The matrix is then attached to the related photo_id. The obtained matrix is then split into training data and test data. The training data counts for 75% of the selected images, while the test data counts for 25% of the images.

2. Cluster model training

To extract the feature, we considered Gaussian Mixture Model and the k-mean clustering. As the feature matrix is quite massive, we decided to use the Mini Batch K-means to reduce the computational cost of clustering. We then fit the training set with the package "sklearn.cluster.MiniBatchKMeans" to get the histogram values with the number of clusters as 20, 50, 100,200, 500, and 2000 for classification. The reason we would like to select numbers of clusters with such a huge range is to prevent the under clustering or over clustering.

3. Classification

Next, we are moving to the classification. We did a Neural network and Logistic regression for the classification. For the Neural network, we first compared the one layer neural network which only has one output layer and 3 layers neural network which has 2 hidden layers. During this process, we did a different size of Epoch to run through the whole data set. We use 5, 10, 20, 50, 100 each time for each Neural network. After we decided to use 3 layers, we ran the different activation function to choose the best. We did 3 different activation function: Sigmoid function, Tan-h function and ReLU function.

For the Logistic regression, we first ran the logistic regression function with default parameters which have default penalty and default C parameter on the different k-means, eg, 20,50,100, 500, 2000 means. The reason we did this step is trying to find the best number of the mean for the algorithm.

4. Evaluation

To get a general idea of how the model worked on 20 image categories, we calculated the confusion matrix for the model achieved the highest accuracy rate. In addition, we used 'sklearn.metrics.classification_report" to build a text report showing the main classification metrics, such as F1 score; precision, and recall.

Results

- Neural Network Model Result

We compared the accuracy rate with the different number of clusters in different epochs in one layer and three layer Neural network. As a result, we found the 500 clusters K-mean has the best accuracy at 5 epoch in three layers Neural network, achieving a 53.3% accuracy rate. The details are shown in Figure 1.

From Figure 2 we can see how the different epochs affect the accuracy. For one layer Neural network, Increase the epochs will significantly increase the accuracy rate. However, for the three layers Neural network, the 5 epoch gave us the highest accuracy rate and increase the epochs decrease the accuracy. It might be caused by the overfitting with too many iterations in the three-layer neural network. We can see the one-layer neural network is much worse than the three-layer network. More layer we get more non-linearity in the network. So we chose the highest accuracy rate which is three layers Neural network with 5 epochs.

For choosing the best activation function, we tried three different activation function: Sigmoid function, Tan-h function, and ReLU function. Sigmoid activation function gives us a better result compared with the other two functions. The reason is that the Sigmoid activation function is good for the multiclass classification task, and tan-h more used classification on the two classes.

Although the Sigmoid function achieved the highest peak point compared with the other two functions, due to the time constraint, our group chose the ReLU as the function to model. With the results above, we are able to get the confusion matrix illustrated in Figure 4. In the figure, the lighter the pixel, the higher the number of pictures is assigned to the related category. We see that the pixels along the diagonal have the lightest color, indicating a relative ideal matching ability of our model.

The main classification metrics for each of the 20 selected landmarks are demonstrated in Figure 5. We see that all three categories follow a similar trend. But they vary a lot with different landmarks. The possible cause of this variation is that some of the landmarks contain vital geometric information, which can be eliminated by SIFT and Mini Batch k-mean algorithms.

- Logistic Regression Model Result

We also tried the logistics regression for classification but we decided to not use it because of the accuracy performance not as good as the neural network. From Figure 6 we can see the logistic regression accuracy rate is increasing at the beginning until it reached around 0.5 at 500 cluster means. But after then, the accuracy rate starts dropping down. And the time it costs is much larger than the three-layer neural network. Since the highest accuracy rate still lower than the Neural network and the time cost, we decided to focus on the Neural network.

- Results comparison

Since we got 53.3% accuracy rate, we tried to compare our results with the winner in Kaggle competition. The winner had 36% accuracy rate for the whole data and we get better result since we had the smaller data.

- Conclusion

From this project, we did a large amount of research to find out the best measure to grab the massive feature matrix obtained from the SIFT algorithm. We found out that by extending the bag-of-words model to classify images instead of text documents, we can achieve high accuracy in image recognition. Within our computation ability, we decided to use the Mini Batch to cluster the features. The measure proved to be robust for the use of the classification by achieving the 53.3% accuracy rate.

In our analysis, we learned that the neural network works better than the logistic regression in terms of the image matching. The reason might be that the complexity of the neural network is larger than that of the logistic regression, resulting in more parameters available to adjust for a better result.

Last but not least, along with the analysis, we realized that SIFT can do a good job in extracting local features, but can also ignore the geometry feature, which might have a large effect on the accuracy of the image matching. We can use WGC to eliminate such a disadvantage, but due to time constraints, such an endeavor is not put into practice.

Supplementary Material

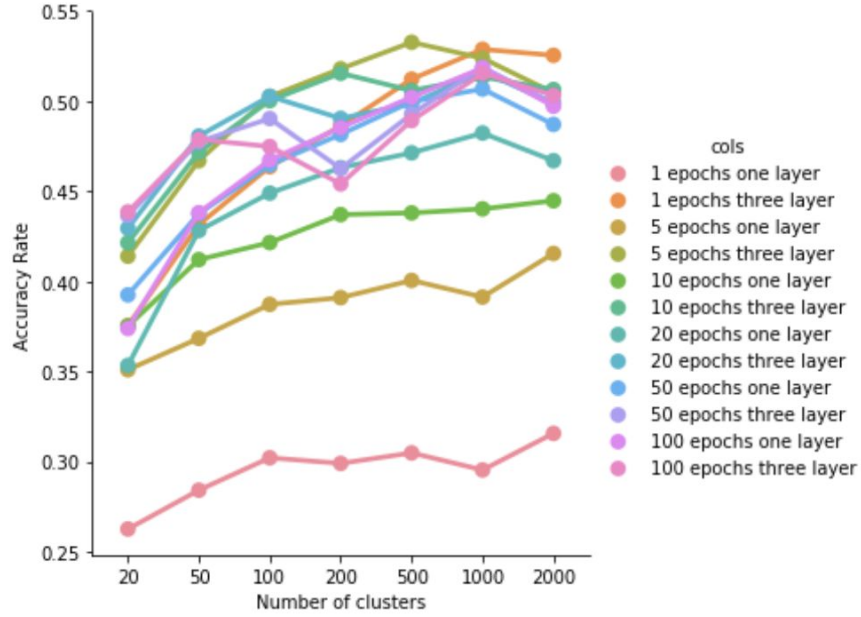

Figure 1: *Neural network result summary*

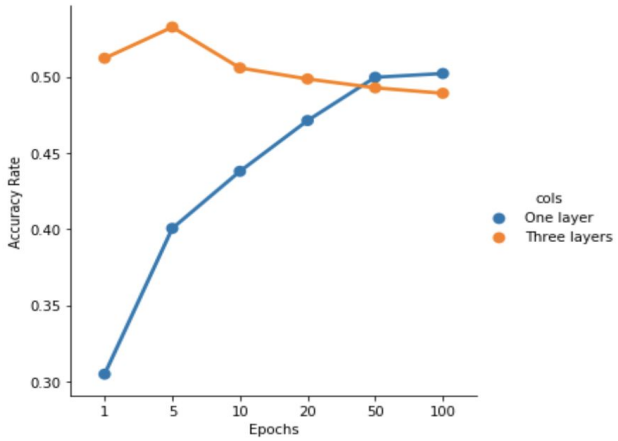

Figure 2*: Accuracy comparison of 500 clusters with different epochs*

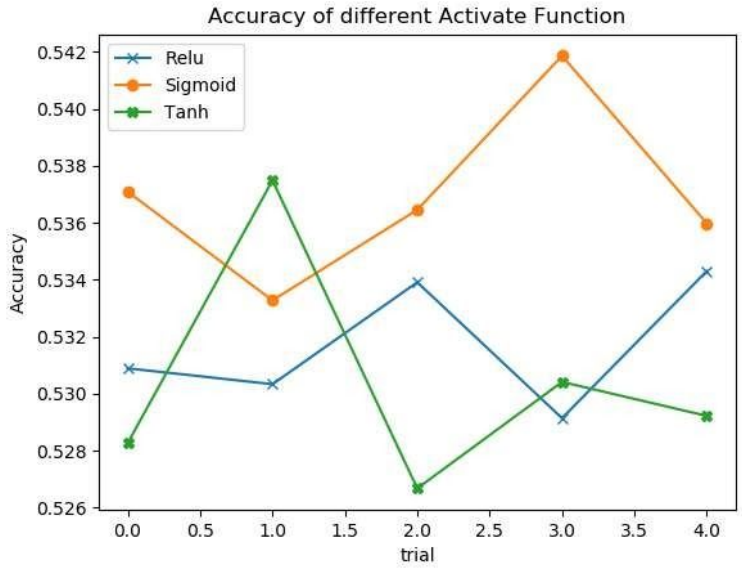

Figure 3: *Accuracy of different active function*

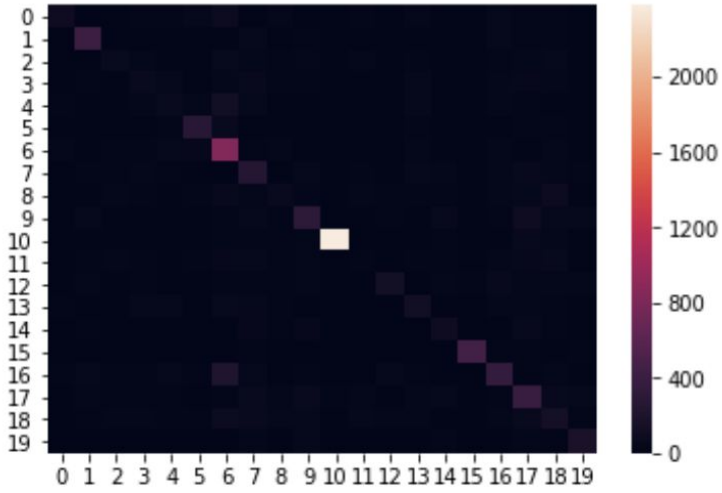

Figure 4: *Confusion matrix of the best fit model*

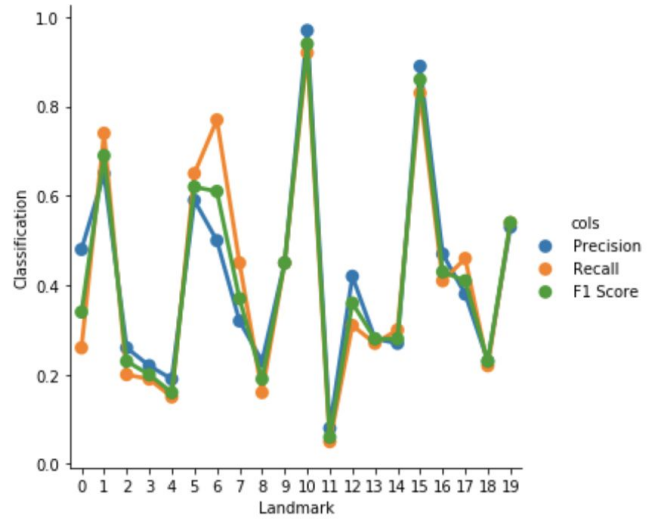

Figure 5: *Classification results for each selected landmark*

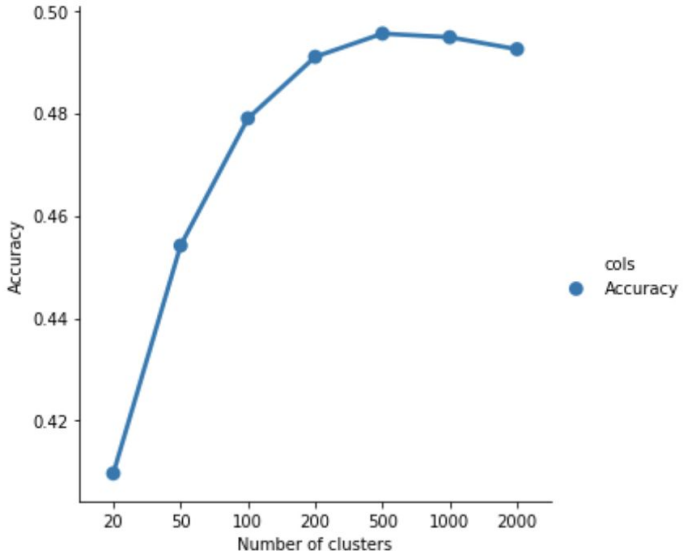

Figure 6*: Logistic regression accuracy rate by the number of clusters*