Landmark Recognition

CPSC 663 Term Project

Abstract

To solve the problem of landmark image recognition, we train a deep convolutional neural network to classify images and learn which travel landmark is portrayed in an image. We find that a transfer learning approach based upon AlexNet is very effective in tackling this problem.

David Adelberg,
Thomas Liao, &
Charles Wong

April 27, 2018

Contents

1 Introduction

The goal of this project is to build a deep learning model that tackles the challenge laid out in the “Google Landmark Recognition and Retrieval Challenge.”

Google sponsored this challenge in order to help users remember their past trips. Classifying different landmarks from around the world will help Google with labelling the image that the user is looking at and with building a natural language interface to its image database.

In order to solve this challenge, we designed, built, and trained an image classification model to effectively perform the recognition task required to a competitive degree.

2 Problem Definition

We seek to train a neural net that will classify an image of a landmark to the appropriate landmark. For example, when the following image of the Leshan Giant Buddha is passed to the neural network, it should output a high probability that the correct class is 9605, which is the landmark id corresponding to the Leshan Giant Buddha.

A photo of the Leshan Giant Buddha

During preprocessing, we resize the input image. So, images X passed to the this classifier lie within $L : \mathbb{R}^{28x28}$. There are 50 classes of landmarks in our dataset, so our function f outputs probabilities $Y \in [0,1]^{50}$. Our objective is to maximize the probability that the correct landmark class is the top landmark class predicted by our network. We felt that this objective is appropriate as there should be only one landmark in an image.

3 Data

To control data storage costs, we used a subset of the over 1 million urls and 15,000 landmark classes provided by the contest organizers to train our network. To generate our subset, we first sorted the landmark classes by frequency. We included landmarks 51-100 in our dataset, for a total of 65,000 images.

To obtain the data, we wrote a script to download these images and resize them to a 28x28 tensor. In total, our training set is 43GB in size and our test set is 5GB in size.

The data we are provided with is in the form of URLs, from which we can write a script to retrieve the actual images. While this project seems similar to ILSVRC in that we are also classifying images (recognition and retrieval), in this project, there are over 15,000 classes (as opposed to 1,000 in ILSVRC), and the number of training examples per class may not be very large. Nevertheless, the data comes from Google images, and the expanded data set contains over 1 million images. In the test data, each image can contain one landmark, no landmark or multiple landmarks.

4 Implementation

For our classifier, we built a deep convolutional neural network. We chose to base our architecture on AlexNet, as we saw in lecture that this architecture has been studied thoroughly in the research literature and has proven effective in a variety of contexts. Our first four layers have the same architecture as the original AlexNet, but TODO CHARLES EXPLAIN ARCHITECTURE.

Inspired by the transfer learning results presented in lecture, we initialized the portion of the network that has the same architecture as AlexNet with the AlexNet weights. Training took place in two stages: in the first stage, we trained the weights in the custom layers of the network. In the second stage, we retrained the original weights. We expected a two-stage approach to produce more robust weights. In addition, we expected that training the AlexNet weights on this new dataset in the second stage could help the first few layers adapt to the problem of landmark recognition.

TODO Charles more details on why we chose the architecture

To tune hyperparameters such as the learning rate for the novel layers and the original AlexNet layers, we trained the model on a small subset of the data. We achieved acceptable performance with a learning rate of 0.001 for the novel layers and 0.0002 for the original AlexNet layers.

In our tests, we found that a L2 regularizer was not particularly helpful, so we did not think it necessary to include one in our code. We think that our network architecture, which has a number of shared parameters in the convolutional layers, served as a form of regularization.

Directions:

Describe your implementation. Include what you chose to use for your solution, the architecture, loss function, regularizations, hyperparameters, and etc. Discuss the reason behind these choices.

5 Results

As discussed in the problem definition, we measure accuracy by TODO DOUBLE CHECK the proportion of test-set images for which the correct landmark class is the same as the class that the network predicts to be the most likely. Our fully trained network achieves a test-set accuracy of X. We are satisfied with this accuracy. Our network predicts the correct class n times for every one time that it makes a mistake. As there are m classes, this suggests that our network performs much better than chance and has learned a lot about classifying landmarks. In addition, this accuracy is high enough to be useful in a business context, such as targeted advertisement sales.

6 Contribution of Team Members

TODO Need to Ensure Accuracy. All should edit to highlight their impact.

All three members of our team played key roles in this project. David led the first stage of the project, setting up the team GitHub and FloydHub and committing the first version of the neural network. Tommy led the second stage of the project, downloading the large dataset, preprocessing it, and uploading it to FloydHub. Charles led the third stage of the project, modifying the initial architecture and training the network.

In addition, all team members participated in architecting and training the network and applying what we learned in the course. The choices we made are the concensus decisions of the team. David led the discussions that spurred us to base our architecture upon AlexNet and to apply transfer learning to this problem. Charles takes primary responsibility for selecting the number of AlexNet layers to customize and for determining the topology of the custom layers. Tommy led the decisions to.

7 Conclusion and Discussion

We are proud of our work on this project and our final result of X accuracy. In addition, this project deepened our knowledge of and familiarity with deep convolutional neural networks. If we had more time, we would have trained our network on the full dataset and submitted our network to Kaggle. The subset of data that we used takes several hours to download using Yale wifi. It would have been interesting to write code so that several CPUs download (in parallel) the terabytes of images in the full dataset. In this case, we would have also trained our network using multiple GPUs. In addition, it would have been interesting to compete in the landmark image retrieval challenge as well. However, we found that working with 50GB of data was more than enough for us to gain valuable experience with deep learning and convolutional neural networks.