Classifying Landmarks with Convolutional Neural Networks

Nikki Hardiman
University of California San Diego
nhardima@ucsd.edu

Abstract

In this project, I classify images from the Google Landmark dataset using a convolutional neural network based off AlexNet. I train the convolutional neural network from scratch, observing the effects of various activation and optimization functions, and increasing the number of layers to obtain a validation accuracy of 75%.

1. Introduction

Convolutional neural networks have become the standard tool for image classification tasks. Because of their extensive flexibility in structure, many successful models have emerged with convolutional neural networks as a fundamental backdrop. AlexNet made a mark in 2012 winning the ILSVRC-2012 competition with its eight-layer CNN [1]. In 2014, VGG made a mark in the ILSVRC-2014 competition for its use of a very deep 19-layer CNN [3]. GoogLeNet also achieved success in 2014 with their 22-layer CNN and utilization of an Inception architecture [2]. Furthermore, ResNet achieved hallmark success in 2015 for their introduction of residual mapping to avoid the degradation problem [4]. In this project, I aim to classify landmark images using CNNs.

In section 2, I outline the Landmark dataset which is used to train and test the model to classify images from landmarks throughout the world. In section 3, I describe the methods to which I utilized in trying to achieve higher validation accuracy and better optimization. In section 4, I outline the experiment conducted. After training the models, I test the model to obtain validation accuracies. I analyze the predictions the model makes to observe how close or far the model is in predicting the correct labels. Finally, I analyze the reasons for mislabeling a few images from the experiments.

2. Google Landmark Dataset

The Google Landmark Dataset was obtained from Kaggle but created by Google and contains images from all around the world. However, from Figure 1, the landmarks mostly originate from Europe. This is supported by the fact that the top two labels containing the most images both originate from Italy. The label 9633, St. Peter's Basilica in Vatican City, contains 49,525 images and the label 6051, the Colosseum in Rome, contains 49,306 images in the dataset. The original dataset contains 1,193,114 images. However, due to memory limitations and computational time, I reduced the dataset size to 58,896 images. I found the top 16 classes, those that contained the most images, and extracted 4000 images per class, or 64,000 images to process in total. The training dataset is comprised of an id for each image, a URL to view the image, and the corresponding integer landmark label. After preprocessing the data, I found 5,104 invalid URLs, producing an overall total of 58,896 images to train and test the model. I normalized the images and split the training and validation to an 80/20 split.

In Figure 2, I display four images from seven classes. Just based on the four images sample, the images range in terms of perspective and angle. Some landmarks have distinguishable characteristics that I hypothesize the model can learn very quickly. Class 6051 and 8429 have discernable buildings and colors: the brown colosseum and a brown narrow tower, respectively. But for other categories some images range in background and angles where it can be hard to distinguish the same label. Class 6651 contains people in the photographs and even includes a nontypical abnormal purple background.

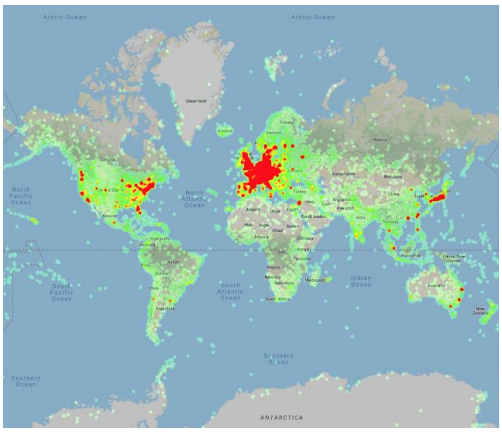

Figure 1. Heatmap of the landmarks from the dataset [6].

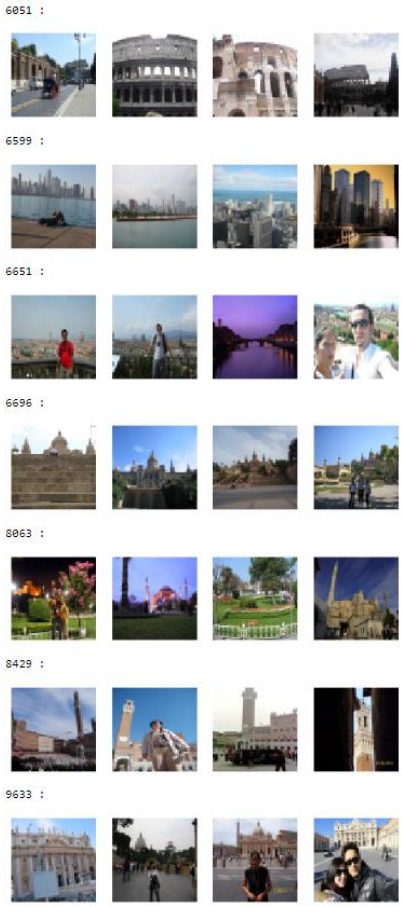

Figure 2. Image samples of the 16 labels from the Google Landmark Dataset.

3. Methods

In this section, I discuss the architecture of my model and features that contribute to the architecture of the model.

3.1. Activation Functions

The most common activation function for most convolutional neural networks has been ReLU [1, 2, 3, 4, 7]. ReLU is widely used because it learns much faster than networks that employ tanh [1]. Other common activation functions that are also utilized are LeakyReLU, ELU, and sigmoid. However, sigmoid and tanh result in much slower learning times due to their saturating outputs as evident in Figure 3. Sigmoid compresses its input into the range [0, 1] and tanh compresses its input to [-1, 1]. Rectifiers tend to have faster learning times and overall better performance as seen in Figure 3 because of its non-saturating output. I decided to use LeakyReLU for my model as Figure 3 shows it has a significant improvement compared to other activation functions.

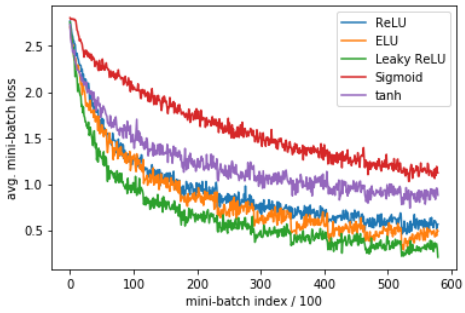

Figure 3. Training loss for various activation functions based on a four-layer CNN.

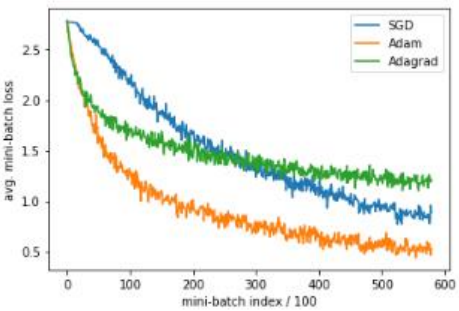

Figure 4. Training loss for various optimizers based on a four-layer CNN. Adagrad decays faster but then stabilizes to a larger value than SGD.

3.2. Optimization

Optimization for neural networks are integral in the learning process for models. Optimizers allow the model to reduce the loss according to the choice of optimization. Stochastic gradient descent has been another widely used optimization choice for CNNs allowing for tuning of momentum and learning rate [2]. Stochastic gradient descent often avoids bad local optima because of its method of stochasticity. Instead of traditional gradient descent, where the gradient eventually converges to any optima, local or global, stochastic gradient descent allows the gradient to jump out of misleading optima usually leading to better performance compared to regular gradient descent.

Another popular optimization technique has been the introduction of Adam. Adam is a stochastic optimization method that only requires first-order gradients and requires little memory [8]. The method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients [8]. Figure 4 shows the difference in the loss function when employing Adam, SGD, and Adagrad, another common optimizer. I utilize Adam in my CNN with a learning rate of 0.001.

3.3. Network Layers

One can observe a common pattern among the leading

convolutional neural networks: increased depth. AlexNet showed its strength in 2012 with an eight-layer CNN and two years later, VGG and GoogLeNet proved CNNs that reached up to 19 and 22 layers achieved greater success in their classification accuracies [1, 2, 3]. While implementing deep CNNs like VGG and GoogLeNet could help achieve higher accuracy, I only compared accuracies among CNNs that ranged from one to five convolutional layers because I was limited in memory and computational space.

Corroborating with the models AlexNet, VGG, and GoogLeNet, I found that CNNs with fewer layers achieved smaller validation accuracies. Table 1 shows the validation accuracy for the varying number of layers for CNNs. Each convolutional block consisted of ReLU and average pooling.

3.4. Overall Architecture

Given that there has been substantial evidence that more convolutional layers results in more significant improvement from the winners of the ILSVRC competition and from Table 1, I used a neural network inspired by the architecture of AlexNet. AlexNet consists of five convolutional networks layers and three fully connected layers, where max pooling follows the first, second, and fifth convolutional layers and ReLU is applied to all convolutional and fully connected layers [1].

My convolutional neural network comprises of five convolutional layers and three fully connected layers where the LeakyReLU non-linearity follows each convolutional layer and fully connected layer. Instead of being applied to the first, second, and fifth layer, max pooling is applied at each convolutional layer.

4. Experiment

I evaluated my model on the modified Google Landmark dataset that contains 16 classes. My model is trained on 47,116 training images and evaluated on 11,780 validation images. I trained the model using Adam optimization with a learning rate of 0.001, and a batch size of eight images. Observing the model's predictions, I then analyzed the differences with and without dropout by looking at the model's predictions for sample images. I then evaluated the model with the validation set and observed the model's accuracy for each class. Overall, my model achieved a validation accuracy of 75% with dropout.

Based on Krizhesky et al., dropout is integral in preventing networks from overfitting. Because dropout sets the output of each hidden neuron with a probability of 0.5, it encourages the model to learn more robust features [1]. I compared my model's predictions with and without dropout. With dropout, my model achieved a decrease in validation accuracy, but seemed to have more accurate

Model	Validation Accuracy
2 CNN	61%
3 CNN	74%
4 CNN	78%
5 CNN	77%

Table 1. Validation accuracies for varying convolutional layers. There is a significant difference between two and three convolutional layers.

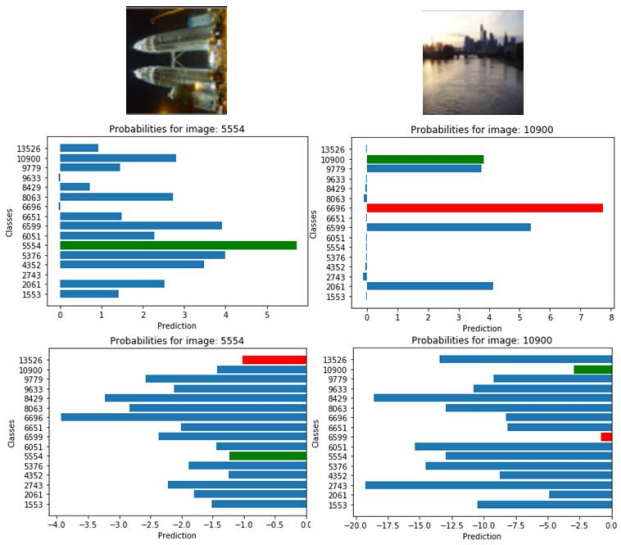

Figure 5. Two images, classes 5554 and 10900, from the validation set. The graphs in the first row correspond to the model that is trained without dropout and the second row corresponds to the model with dropout in the first two fully connected layers. Green indicates the correct choice and red indicates the incorrect prediction the model made.

predictions observed from some sample images in Figure 5. Without dropout, the model obtained a validation accuracy of 77%. Figure 5 compares the model's predictions for several images with and without dropout.

Although dropout has been proved in improving model performance by preventing overfitting, I found it surprising that without dropout, the model achieved a higher validation accuracy. ResNet opted out of utilizing dropout and took first place in the ILSVRC 2015 classification task [4]. There appears to be considerable evidence for and against dropout. However, based on Figure 4 it is interesting to note that without dropout the model tended to have larger differences in its errors while the model with dropout had a more subtle difference between the incorrectly predicted label and the correct label. This observation seems to suggest that the model is learning more robust features. Figure 4 shows that predictions without dropout seem to have a larger choice of potential correct options while predictions with dropout contain a more condensed pool of potential correct

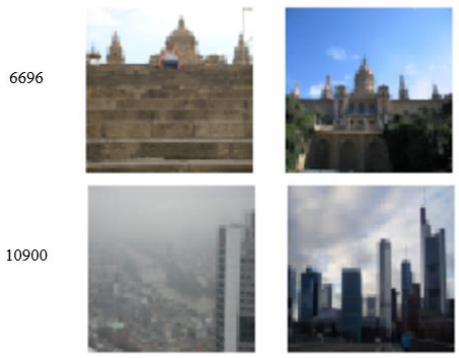

Figure 6. The first row of images from class 6966 resulted in the largest accuracy of 90% while the second row of images for class 10900 resulted in the smallest accuracy of 40%.

options. The model that predicted incorrectly for the class 10900 without dropout has a correct label that is relatively equal to other class labels such as 9779, 6599, and 2061. But with dropout, the model's prediction value for 10900 has a smaller difference between the incorrect prediction. This implies that the model is learning more of the features and is condensing the choice of correct predictions to be more precise.

I only tested the model on its validation accuracy with 16 classes because the original dataset had 14,947 classes in total. Most classes contained less than 100 images. Furthermore, the model contained large disparities in accuracy among each class. Several classes obtained a validation accuracy above 85% while other classes as low as 66% and even 40%. In Figure 6, the images for 6696 appear to have a distinguishable and prominent structure of the same color despite backgrounds of white and blue while the images for 10900 seem to range drastically in content yet are labelled under the same class. The images seem to belong to different landmarks; there is no distinguishing structure like the landmark for 6696. This highly diverse set of images for 10900 seem to corroborate the low rate of accuracy the model output.

5. Conclusion

In conclusion, I observed the results of a convolutional neural network inspired by the AlexNet architecture trained on landmark images from Google. I compared the efficiencies of various activation and optimization functions, finding that LeakyReLU and Adam outperformed their counterparts. I also compared the accuracy of convolutional neural networks that varied in layer size, discovering more layers improves model performance. I then compared the differences in results with and without dropout by analyzing the predictions for each class for two images. Training the model from scratch resulted in a validation accuracy of 75% for the Landmark dataset.

Because computational memory was limited, future experiments could use a larger Landmark dataset that included more than 16 classes and experiment with increasing the number of layers to achieve higher model accuracy.

References

[1] A. Krizhesky, I. Sutskever, and G. Hinton. Imagenet classification wth deep convolutional neural networks. In NIPS, 2012.

[2] C. Szegedy, W. Liu, Y. Jia, et al. Going Deeper with Convolutions. *ArXiv e-prints*, Sep. 2014.

[3] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Image Recognition. *ArXiv e-prints*, Sep. 2014.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *ArXiv e-prints*, Dec. 2015.

[5] Google Landmark Dataset. https://www.kaggle.com/google/google-landmarks-dataset

[6] Google Landmark Dataset Description. https://ai.googleblog.com/2019/05/announcing-google-landmarks-v2-improved.html

[7] M. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. *arXiv e-prints*, Nov. 2013.

[8] D. Kingma and J. Lei Ba. Adam: A Method for Stochastic Optimization. *arXiv, e-prints.* Dec. 2014.