Comparing Pre-Trained Convolution Neural Networks on the CIFAR-10 Dataset

Archibald Latham

Abstract

Convolution Neural Networks or CNNs are a relatively recent but very promising development of computer science. In this paper, I discuss the implementation of several pre-trained CNNs on the CIFAR-10 dataset. The CNNs that I evaluate are, ShuffleNet, GoogLeNet, and ResNet18. These are well respected and detailed image classification CNNs that have been extensively developed over the past decade thanks to the ImageNet Large Scale Visual Recognition Challenge or ILSVRC. The code used to carry out the analysis of these networks is open source and utilizes PyTorch alongside CUDA. To evaluate each CNN I recorded its accuracy and loss over 10 epochs for both training and testing data. To analyze my results I plotted these data points against each other across epochs. These results and a discussion of each network can be found in the paper below. Overall CNNs are a revolutionary tool in the realm of computer vision and having an understanding of how they work and their improvement over time is a crucial step to developing them further.

Introduction

CNN's function by taking a set of inputs, in our case images, and convolving these images across several filters. The filters within the system vary and allow the system to detect the prevalence of certain image features such as differently oriented edges.[2] This process takes place across several layers of a system before a prediction is made. This prediction is given by showing the correlation between the predetermined labels and the identified features. To make meaningful predictions though the system must be trained with several thousand training images that can be fed to the system over several epochs. Epochs signify the training of a system over a full dataset and training typically occurs over several epochs that are divided into batches.[6] In this process of training the network takes each image and makes a guess on it's label. The system is then given the correct answer and uses its error to correct itself through back propagation. In this process the value of certain features are given more or less weight in the final prediction.[3] Once a model is trained it can be fed a testing dataset that evaluates its accuracy and efficiency. The accuracy is evaluated by percentage of correct guesses across a set of images and a loss function that measures the effectiveness of backpropagation. Efficiency is measured by evaluating how these attributes improve across epochs.

Related Works

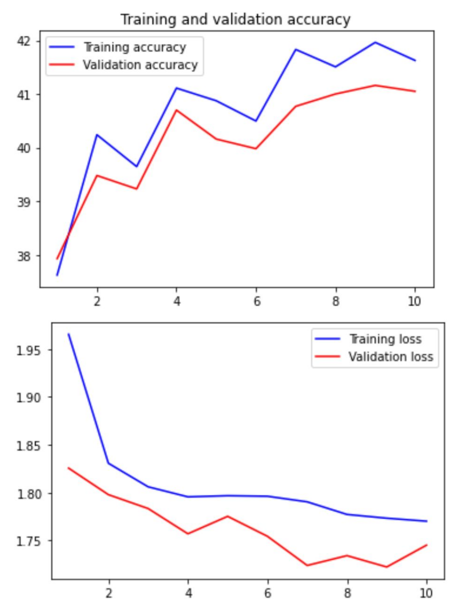
In terms of background, numerous papers have been written on each of these systems and how their respective structure is effective for image classification. GoogLeNet was developed by Google in 2014 and is based on a system called LeNet. The network also goes under the name Inception v1 as the system has been further developed into improved versions v2, v3, and v4. The system has 22 layers in total and won the ILSVRC in 2014.[7] ResNet18 on the other hand is a competing system developed in 2015 by Microsoft and Facebook that won the ILSVRC competition the following year in 2015. The network has 18 layers but has been expanded to larger versions such as ResNet34.[4] SqueezeNet is a CNN developed in 2016 by the company DeepScale alongside UC Berkeley and Stanford. The network is based off of AlexNet, a network developed in 2012 that won the ILSVRC competition for that year. Squeeze net boasts a system that is as accurate as AlexNet with a significantly smaller size.[5] Overall each of these systems has highlighted a major milestone in the development of image classification CNNs. While many papers have been written on the effectiveness of each of these systems, few papers have compared these systems against each other. My goal in this paper is to evaluate how these systems have held up against each other over time.

Method

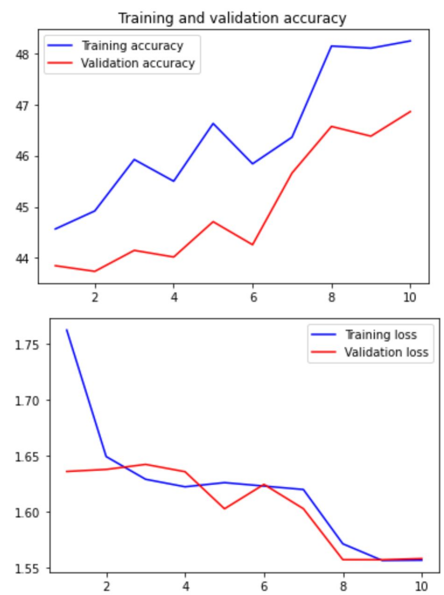
The code used in this project to test the CNNs is a modified version of open source code provided by Christian Carpenter. The code utilizes PyTorch, a machine learning network for Python, alongside CUDA, a graphic processing unit or GPU interface.[1] The CNNs that I evaluated were, SqueezeNet, GoogLeNet, and ResNet18. The reasoning for choosing these three systems is that they all differ in structure and have each competed in the ILSVRC. The training and testing data used in the analysis is the CIFAR-10 dataset. This dataset is chosen for it’s accessibility and common use. To evaluate the CNNs, I trained and tested each system over 10 epochs and recorded the accuracy and loss functions over both the training sets and test sets. To analyze the maximization of the accuracy and minimization of the loss I plotted these values against each other over epochs. The results and their analysis can be found in the sections below.

Results

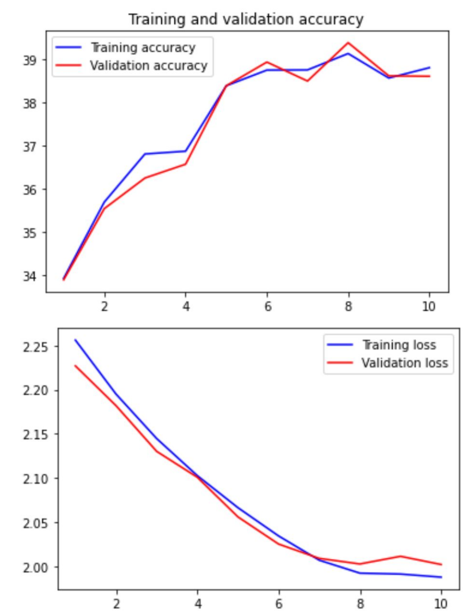
GoogLeNet(2014):



ResNet(2015):



SqueezeNet(2016):



Discussion:

What we can see from these results is that progress has been made on the development of CNNs overtime in several significant ways. The first thing to note is that the achieved accuracy of ResNet is higher than GoogleNet and that's it's loss function minimized to a lower value faster. This shows significant improvement by these winners of the ILSVRC across only one year. Another interesting thing to note is that while the overall accuracy and loss of SqueezeNet is worse than GoogLeNet and ResNet that it's difference between these values for the training and validation are significantly smaller. This indicates a much more efficient and succinct system. Considering that SqueezeNet has the same accuracy as the system AlexNet that won the ILSVRC in 2012 we can predict that in the coming years we will start to see systems that meet the accuracy of GoogLeNet and ResNet with more efficiency. Overall the further development of these image recognition systems is extremely important and highlights a very exciting period in the history of computer vision.

References:

[1] C. Carpenter, "ccarpenterg/LearningPyTorch1.x," *GitHub*, 14-Jan-2020. [Online]. Available: https://github.com/ccarpenterg/LearningPyTorch1.x. [Accessed: 30-Apr-2020].

[2] S. Das, "CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more," *Medium*, 17-Sep-2019. [Online]. Available: https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5. [Accessed: 30-Apr-2020].

[3] S. Gollapudi, "Deep Learning for Computer Vision," *Learn Computer Vision Using OpenCV*, pp. 51–69, 2019.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] Iandola, Forrest, M. W., Ashraf, Khalid, Dally, W. J., Keutzer, and Kurt, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," *arXiv.org*, 04-Nov-2016. [Online]. Available: https://arxiv.org/abs/1602.07360. [Accessed: 30-Apr-2020].

[6] N. Milosevic, "Convolutions and Convolutional Neural Networks," *Introduction to Convolutional Neural Networks*, 2020.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.