Leslie Juengst

Final Project Individual Report

Introduction

The purpose of this project was to explore the use of the Caffe framework to perform a binary classification of dog and cat photos utilizing two different types of networks. Briefly, the LeNet CNN (Lecun et al., 1998) consists of 2 convolutional layers each with max pooling followed by 2 fully connected layers with ReLU transfer functions. The AlexNet architecture (Krizhevsky et al., 2012) contains 5 convolution layers and 3 fully connected layers along with 2 dropout layers and ReLU transfer functions on all layers. The data for this project was retrieved from Kaggle (dr. Avicenna, 2019). The discussion of the project goals, strategies and implementation of the network was generally shared.

Individual Work

Caffe can read several different data structures, and we opted for lmdb as our input data format. It was my responsibility to debug the issues we were having in creating our database from the jpeg files downloaded from Kaggle. Ultimately, I realized that there were specific files that the code was having trouble reading and we deleted those files. Secondly, I researched the image processing that typically occurs prior to or during the training of a network designed for image data. The create_lmdb_tutorial.py file that was written by Hagan et al. and obtained from the github repository of Dr. Amir Jafari, contains a snippet of code that increases contrast in the image and resizes it to a user defined shape. I discovered that it was also common to subtract the mean value of each pixel from the data in order to ensure zero mean, and that Caffe can handle the processing if provided with the binary.proto file of the mean image value. I located the tool that comes with the Caffe package that creates the image mean and generated it then modified the architecture .prototxt file to contain data processing parameters including “mirror” and “mean_file.” Poor documentation meant that took considerable time. Next, following the retrieval of a basic LeNet prototxt file, obtained from exam 2 files, the majority of effort was spent playing with various parameters in an effort to discover the issue behind some strange results. Training was performed a handful of times, and generally resulted in one of two outputs. Figure 1 below shows test accuracy and training loss for the first training run. Decreasing training losses coupled with decreasing accuracy scores presents a conundrum. Figure 2 below shows the second, more common, output where test accuracy was immediately 100%, training losses went from low to high and remained contained through any number of iterations. I thought perhaps it was an overfitting issue and added dropout layers. I also modified various parameters such the learning rate, the learning rate policy, the stepsize of an altered learning rate policy all of which had no impact. Next I started making more significant changes such as changing the kernel sizes then removing one of the convolution layers. Even then, there was no change and further the value of the stagnant loss remained the same. This made me question the data, and I reviewed the structure and input of the data to the network as well as checked the jpeg files for obviously erroneous data such as negative values. I spent a good deal of time researching this and trying various parameter changes and have yet to figure this out. Sam suggested changes to the gamma value and that helped at least produce more reasonable outputs if not correct. Finally, the AlexNet implementation was my responsibility and I was able to produce a network though the accuracy and loss traces again make the validity of the model circumspect. The accuracy merely oscillates between zero and one and the losses do not seem to decline.

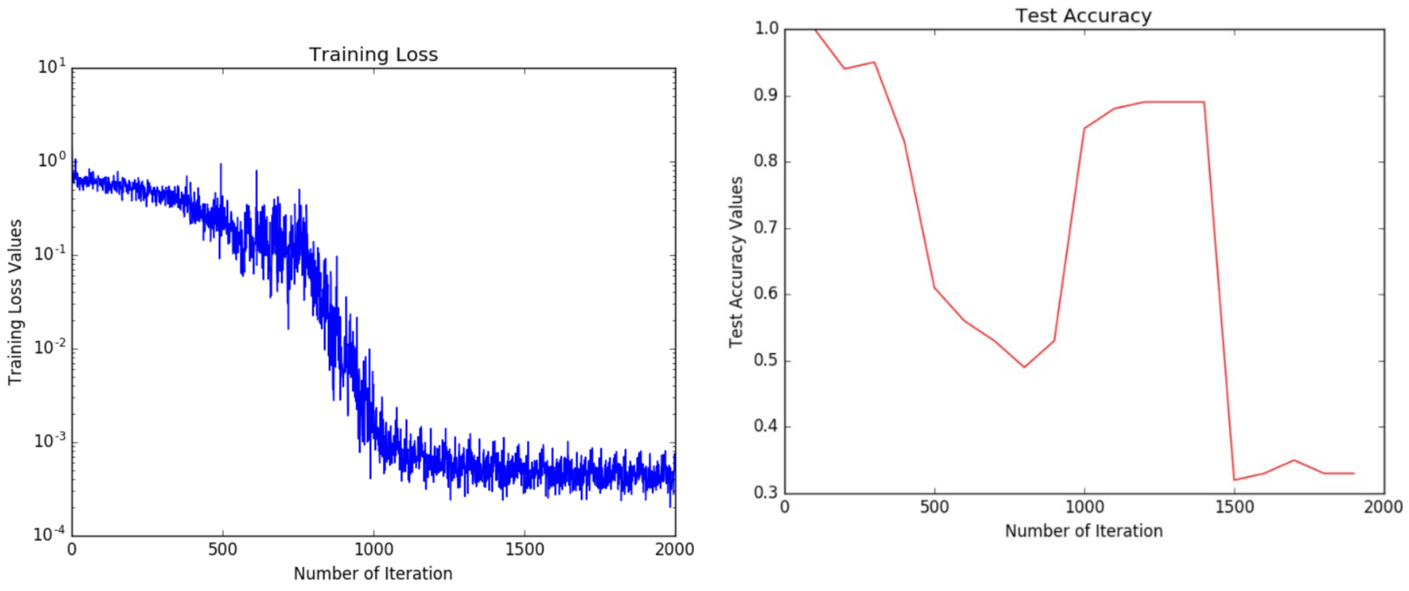


Figure 1: Training loss and Test accuracy of first LeNet CNN of Cats and Dogs dataset

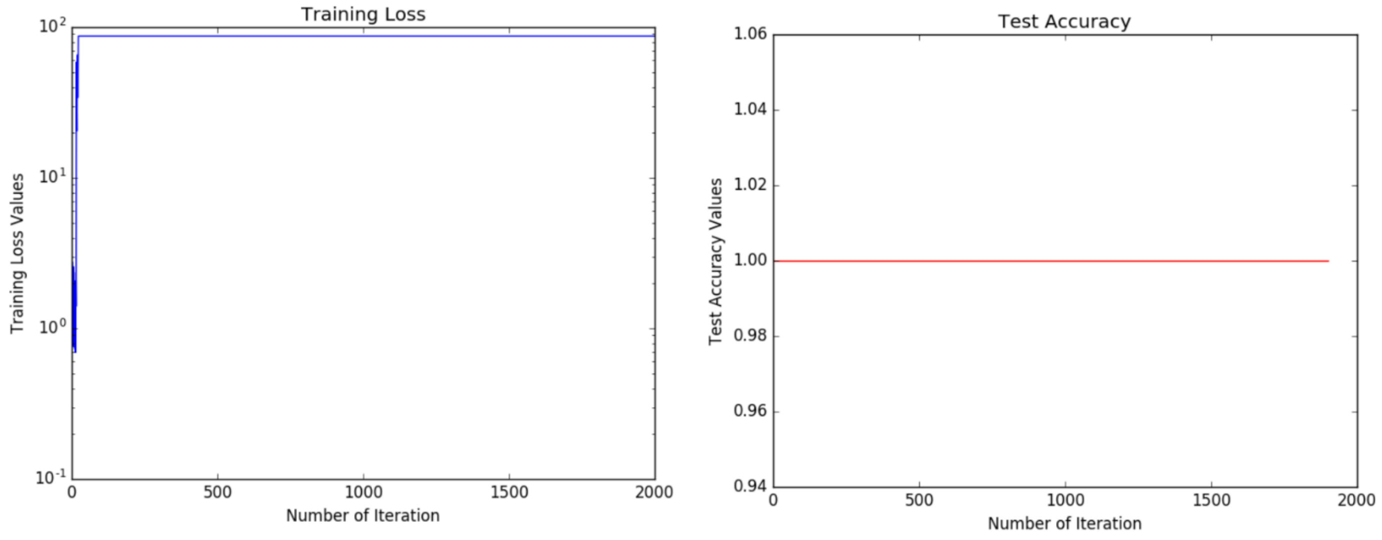
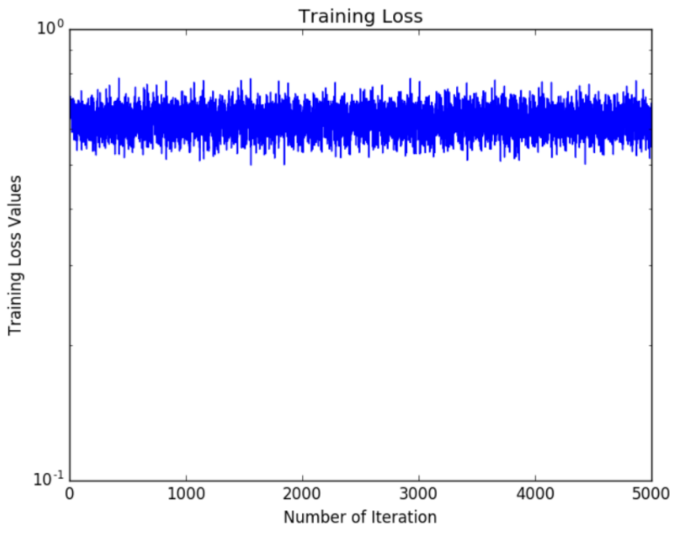


Figure 2: Training loss and Test accuracy of first LeNet CNN of Cats and Dogs dataset

Results

The networks we attempted did not perform well despite various attempts at parameter tuning. The LeNet should have done very well in consideration of the size of the dataset in particular. Instead, the losses remained constant and the accuracy scores are unrealistic at best. Figure 3 below shows the outputs of the AlexNet demonstrating the obvious instability and lack of utility of this model. The suggestion of increasing gamma values improved the situation only very slightly. As gamma increases, the speed with which the learning rate decreases as iterations move forward increases. This value should not have dramatically impacted the results and created a more normal looking loss function and accuracy. Overfitting could perhaps be the problem with such as small data set. In addition, both the training and testing data sets are imbalanced in terms of labels and could stand to be either supplemented or oversampled.



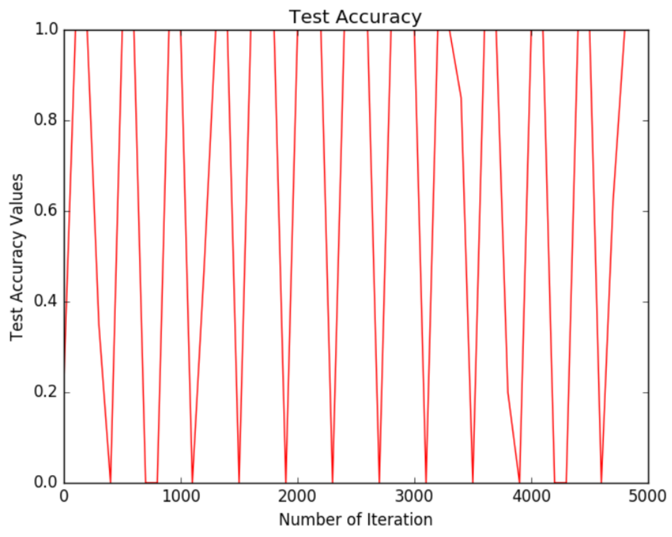


Figure 3: Training loss and Test accuracy of the AlexNet CNN of Cats and Dogs dataset

Summary and Conclusions

The training of a convolution network to identify images of cats and dogs was unsuccessful. The outputs imply that there may be an issue of overfitting or a more basic technical issue that is as yet unidentified. I learned something of the image processing that takes place prior to this kind of model building and a good deal about the number of parameters that can be modified in order to obtain a result.

Code

Code	**% Borrowed**
create_lmdb_tutorial.py	100
alexnet_train_test_oxford.prototxt	97
lenet_solver_oxford.prototxt	99
lenet_train_test_oxford.prototxt	100

References

dr. Avicenna. Cats and Dogs Breeds Classification Oxford Dataset, 2019. URL https://www.kaggle.com/zippyz/cats-and-dogs-breeds-classification-oxford-dataset#annotations.tar.gz. 2019.

M. Hagan, A. Jafari, S. Wang. Create lmdb tutorial. URL https://github.com/amir-jafari/Deep-Learning/blob/master/Caffe_/3-Create_LMDB/create_lmdb_tutorial.py

A. Krizhevsky, I. Sutskever, and G.E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25 (NIPS)*, 2012.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *the proceedings of the IEEE*, Nov., 1998.