Julian Singkham
CS 2300/11
Dr. Berisha

Lab 7: Digits

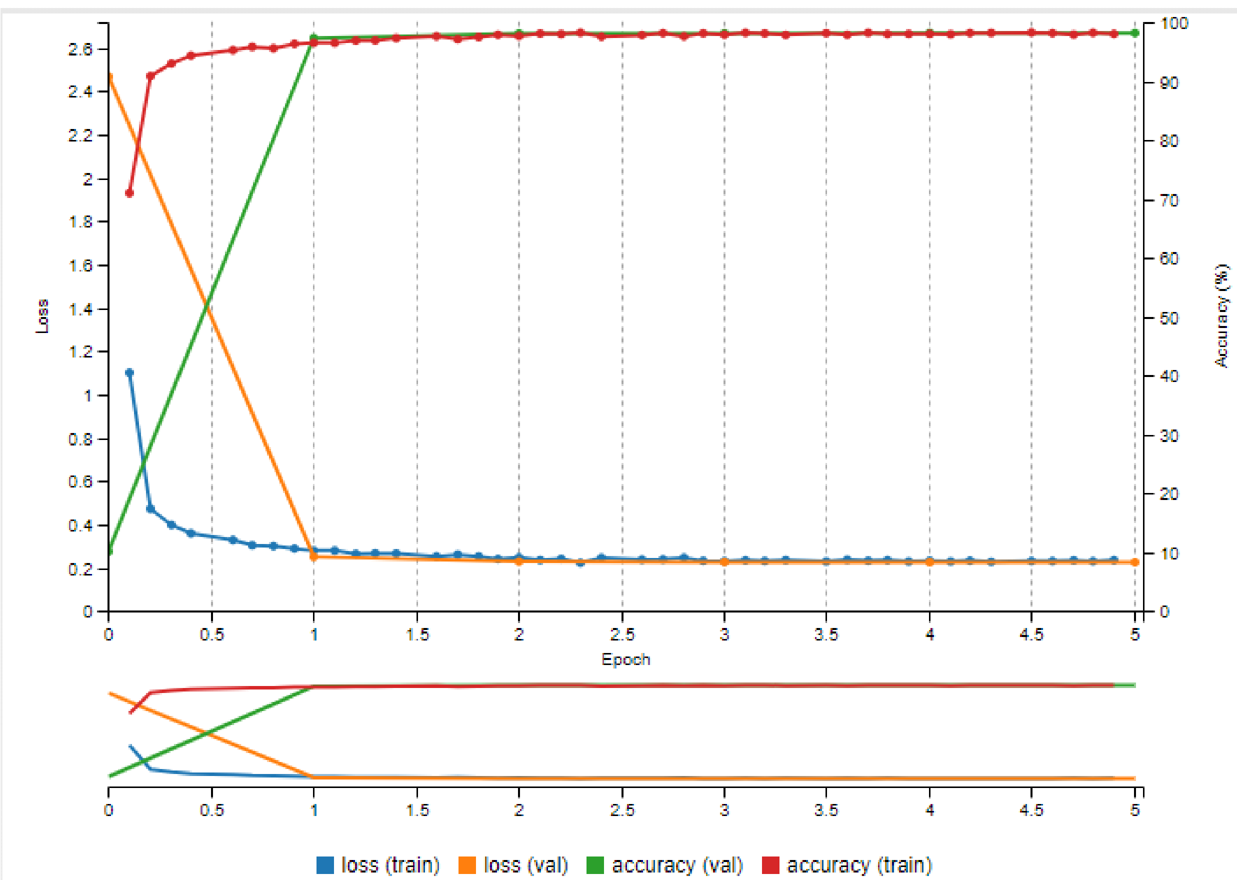
Introduction

The purpose of this lab was to familiarize ourselves with training a Deep Neural Network using DIGITS and the DIGITS software. Adjustments were made to the training algorithm to learn how it affects the model and statistics were taken to benchmark the effects of the adjustments.

Part 1: Data

The MNIST data set contains 70,000 examples of hand written digits; 60,000 for training and 10,000 for testing. Additionally the dataset has been normalized, meaning all the images are of the same size and centered. The data set is a subset of the much larger NIST data set, and contains two parts of the NIST data set. The first half is composed of hand written digits from high schoolers and the second half is from the Census Bureau employees. This approach of diversifying the data lowers the chance of biased data.

Part 2: Train an Initial Model



Part 3: Inference

Predictions

2	99.99%
0	0.0%
1	0.0%
8	0.0%
6	0.0%

Using /data/mnist/test/2/00001.png, the model correctly predicted the number 2 which is not surprising as 2 is a very distinct number since the number 2 is the only number with a horizontal line at the bottom from left to right.

Predictions

1	22.06%
5	17.83%
7	17.3%
9	9.21%
2	9.02%

Predictions

5	88.41%
8	8.5%
7	1.24%
2	0.89%
0	0.63%

Given a number 5 picture (referenced as 5a), the first iteration was unsuccessful as it predicted the image was 1 at 22.06% certainty and 5 at 17.83%. My initial thought was that the image was not sized correctly as the model was trained on 28x28 pixel images and the one I used was 410x402. I believed the vast difference in size caused issues with the model not being able to handle that much white space and incorrectly cropping the image. Once I cropped and downsize the image to 28x28 (referenced as 5b), the predication was 5 at 88.41%. This augmentation solved the issue of the initial incorrect prediction.

Part 4: Epochs, Batch Size

With an epoch of 1, number 2 prediction lowered to 99.47%. The number 5a image predicted 1 at 41.3% and 5 at 17.44%. The number 5b image predicted 7 at 46.66% and 5 at 31.6%. Overall an epoch of 1 decreased the accuracy and prediction confidence. This is most likely due to only running the model in one epoch. Having the model run multiple times allow for reinforcement training as the model can compare its current knowledge with the training data set and derive a more accurate understanding of the data.

Batch sizes 128, 256, and 512 were used. The prediction confidence dropped for number 2 as batch sized increased (97.50%, 88.78%, 67.72%). Number 5a confidence is inconclusive as it increased for all batch sizes except 256 (25.36%, 20.18%, 26.6%). Number 5b confidence rapidly decreased at batch size 128, but quickly grew as batch size increased. (19.15%, 40.97%, 40.78%). As expected, increasing the batch size decreased accuracy as the model did not undergo enough revisions to acquire an accurate knowledge of what a number looks like.

Overall, the GPU utilization percentage did not change. I believe the lack of GPU utilization changing is the fact that the time it takes for the model to re-evaluate its parameters between batches is practically negligible. If the timing was significant, I would see drops in utilization corresponding to when the model is finished with a batch. Additionally, batches are conducted in a sequential fashion, therefore parallelization cannot but utilized to split up the batches between all the GPU cores.

Part 5: Data Augmentation

<u>Vertical</u>

Number 2 confidence decreased slightly to 98.83%. Number 5a drastically increased to 62.64%. Number 5b drastically decreased to 6.92%.

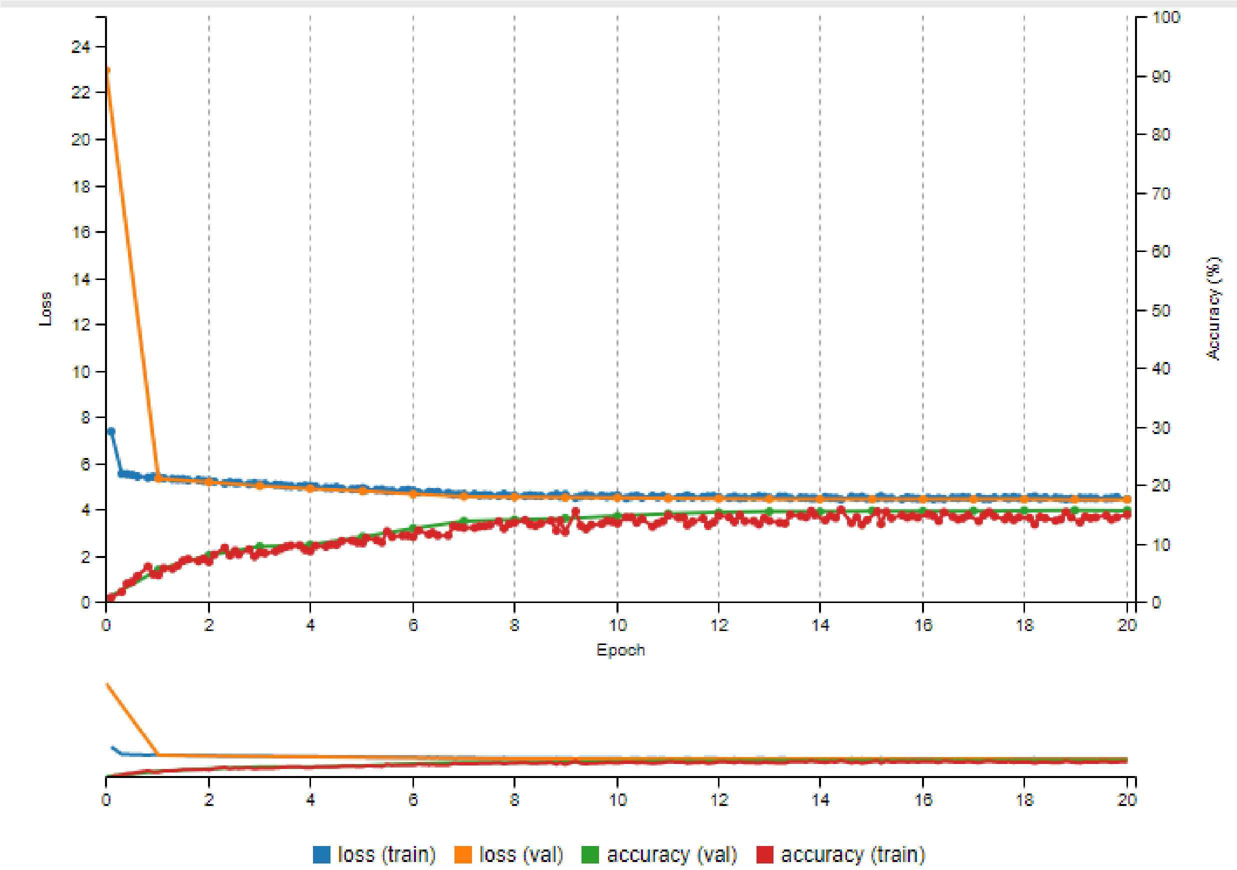
<u>Horizontal</u>

Number 2 confidence slightly decreased to 98.89%. Number 5a drastically increased to 57.25%. Number 5b slightly decreased to 66.72%.

Comparing horizontal to vertical augmentation, horizontal appears to be the best augmentation as that made the model correctly predict all three numbers used. Furthermore, horizontal augmentation gives the model a much higher rate of accuracy when compared to non-augmented. This is to be expected as the possibility of someone writing upside down is far higher than if someone were to write mirrored. Writing upside down involves flipping the paper whereas mirrored writing requires skill and, for the most part, is not done accidentally. Additionally, if there were errors in the dataset, rotating the paper is a plausible error whereas there is no way to manipulate paper to show mirrored text.

Part 6: Another dataset

No data augmentation was applied to the Caltech256 data set as the images used vary greatly in amount of detail, color, shading, etc. I hypothesize that with the current knowledge I have, any pre-processing would negatively impact accuracy. Based on the data from Joshua Goldshteyn, it would appear my hypothesis is some truth in it as he only achieved an accuracy of 14%, whereas my model has an accuracy of 16.35%.



Parameters wise I only adjusted the epoch size to be 20, as consecutive iterations should raise the accuracy level. This is proven somewhat true by accuracy increasing at a logarithmic with rate with epoch. In terms of overfitting, I believe it would occur at epoch 12 due to the very sharp decrease in the rate of accuracy increasing. This change in slope would mean the Model is unable to advance any further and further iterations would reinforce bad labeling.

Predictions

012.binoculars	4.43%
019.boxing-glove	3.96%
005.baseball-glove	3.45%
223.top-hat	2.36%
061.dumb-bell	2.09%

Predictions

191.sneaker	2.07%
157.pci-card	2.05%
138.mattress	1.96%
232.t-shirt	1.87%
238.video-projector	1.86%

Given the two image classifications, /data/caltech256/002.american-flag/002_0001.jpg (flag) and /data/caltech256/010.beer-mug/010_0008.jpg (mug), we can see how the Caltech model is highly lacking in accuracy as the suggested labels are extremely different than what the image actually is.

In general, my model took 15 minutes and 17 seconds to compute, with the GPU averaging 65-70% usage. This lab was very interesting as this is the first time I'm able to utilize a machine learning algorithm and begin to understand the process involved with training a model. The Caltech data set taught how complicated complex objects are to a machine. What we think is easily identifiable is quite difficult for a machine to understand. Its also quite funny to see what the model believes an image is, take for example the mug. None of the proposed objects remotely resemble or are made of the same material as the mug.