CSCI420 Final Project

Daniel Raw*
dsraw@email.wm.edu
College of William and Mary
Williamsburg, Virginia, USA

Abstract

For this final project in Dr. White's Neural Networks for Machine Learning class, we explore supervised, self-supervised, and unsupervised learning, covering a few modalities (vision, text, audition). Conceptually, this project exercised my understanding of important machine learning concepts such as input representation, error functions, optimization methods, regularization, and validation. Practically, this project gave me some experience with popular frameworks such as TensorFlow and PyTorch.

Keywords: convolutional neural networks, image analysis, residual learning, transformers, text analysis, variational autoencoder

ACM Reference Format:
Daniel Raw. 2021. CSCI420 Final Project. In *William and Mary '21: ACM Symposium on Neural Gaze Detection, May 15, 2021, Williamsburg, VA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/1122445.1122456

1 Part I: Convolutional Neural Networks for Image Analysis

1.1 Context

Part I of this project was based off of the paper, *Deep Residual Learning for Image Recognition.* The paper presents a residual training framework to ease the training networks that are substantially deeper than those used previously. The authors of this paper: He, Zhang, Ren, and Sun explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. The authors provide comprehensive empirical evidence proving that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. In this project we try to replicate their findings.

In my research, we critically assess 50-layer and 101-layer residual networks on the same CIFAR-10 image data, undergo an exploratory analysis on the data set, and conduct a comprehensive statistical analysis with data visualization. Similar to the paper, our objective function: the categorical cross entropy loss function is optimized with stochastic gradient descent.

1.2 Exploratory Data Analysis

My exploratory analysis can be surmised into four different analyses. For my first analysis, I looked at the R, G, B channels of one image from the training and testing sets, and tracked the minimum, maximum, and average intensities for each channel. To calculate these intensities, I scanned through 60,000 images from the CIFAR-10 data set and recorded their minimums, maximums, and averages. From my findings, I found that the maximum intensities were 255 for each channel which comes to no surprise since it is highly possible for a pixel to have a maximum RGB value. On the other hand, the minimum intensities ranged from 20 to 23. The average intensities for both the B and R channels was about 125, while the G channel's average intensities was about 100; however, all of the RBG channels all had a frequency of around 1600.

For my next analyses I took a sample image and distorted it by setting certain regions of the picture to have maximum intensities of 255 of different RGB channels. From my observation, when a full intensity of a color channel is put over a white pixel it has lesser significance than when a full intensity is put over a colored pixel. When I put a red channel over a white pixel there is almost little to no change in the shading. This is because the color white is created by using a maximum color intensity of red, green, and blue.

For my next analyses, I took a sample image from the training set and plotted this image three times using one RGB color channel for each. From this simulation it became very obvious that the blue color channel is responsible for darker shades and values as the image using the blue channel had a significantly darker shade than the rest of the images, making it difficult to detect features that were in the original sample image. On the other hand, the image using the green channel had the brightest depiction, making it the easiest to identify features from the original sample image. The image using the red channel was darker than the image using the green channel, but brighter than the image using the red

*This project was completed with a collaborative effort with the 11 other students enrolled in CSCI420 and CSCI520.

Williamsburg '21, May 16, 2021, Williamsburg, VA
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/10.1145/1122445.1122456

channel; however, it was very easy to pin point features from the original sample image.

For my final analyses, I compared the average RGB color intensities across 10 classes from the CIFAR-10 training and validation sets. To easily compare the intensities, I created a bar graph that depicts the intensities of each color channel for each class. For each RGB channel, red had the highest average with an intensity of at least 115 for each class and had an average intensity of 127.44. Blue had the second highest average with an intensity of at least 110 for each class and had an average intensity of 123.97. Finally, green had the lowest average with an intensity of at least 90 for each class and had an average intensity of 122.22.

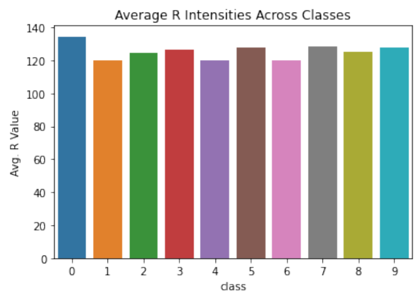


Figure 1. Mean R Intensities Per Class

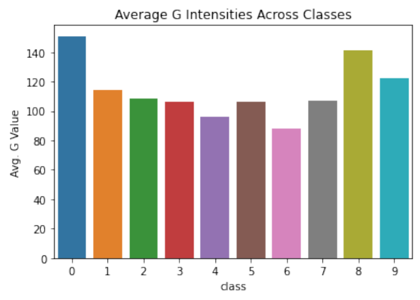


Figure 2. Mean G Intensities Per Class

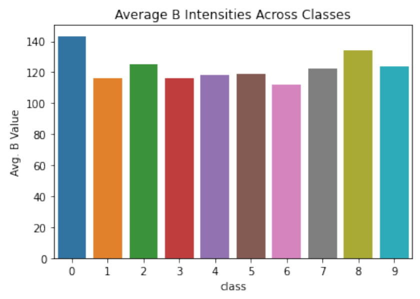


Figure 3. Mean B Intensities Per Class

1.3 Reproduced Research Results

To reproduce the research results, I implemented three functions to record the data from our training model and then plotted it to recreate figure 4. To begin, I instantiated ResNet-101 and ResNet-50 using the keras model function. ResNet-101 is a deep residual model with 101 layers and ResNet-50 is a deep residual model with 50 layers. Next, for both models, I preprocessed the input data for the training and validation sets and set epochs to 20 and batch size to 128. As previously mentioned above, I set categorical cross entropy as the cost function, optimized it with stochastic gradient descent, and used a softmax to add the dense layer. For time purposes, I did not pretrain the layers. For each iteration, I recorded: the top-5 categorical accuracy for the training and validation sets, the accuracy percentage, and the recorded loss.

Figure 4 illustrates the error percentage of ResNet-50 and ResNet-101 from the CIFAR-10 training and validation sets. Surprisingly, ResNet-50 had a lower error percentage than ResNet-101. Figure 5 displays the loss behavior for both ResNet-50 and ResNet-101, and the results in loss behavior also reveals that ResNet-50 performed better than ResNet-101. When looking at the top 1 and top 5 error rates on the training and validation data sets, it further attests that ResNet-50 has a lower error percentage than ResNet-101. Therefore, this means that my implementation did not address the degradation problem even though the solution space of the ResNet-50 network is a subspace of the ResNet-101 network. In theory, the accuracy should increase when there is more depth. A reason for this phenomenom could be because I did not pretrain the weights, or because of my outdated software.

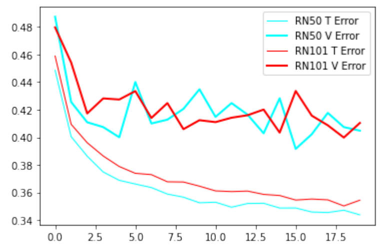


Figure 4. Training on CIFAR-10. The thin line represents the training error and the thick lines represent the validation error.

2 Part II: Transformers for Text Analysis

2.1 Context

Part II of this project was based off the paper, *Attention is All you Need*. The authors of this paper: Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, and Kaiser present a new simple network architecture, the Transformer, that is based solely on attention mechanisms, and dispenses with recurrence and convolutions entirely. The models presented in this

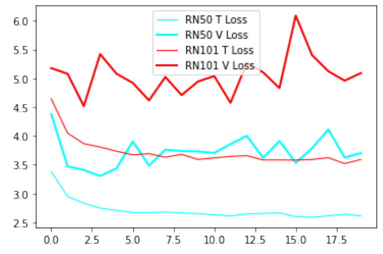
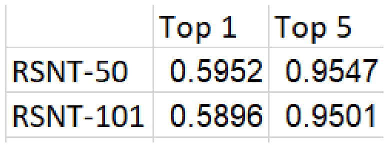


Figure 5. Training on CIFAR-10. The thin line represents training error and the thick lines represent the validation error.



	Top 1	Top 5
RSNT-50	0.5952	0.9547
RSNT-101	0.5896	0.9501

Figure 6. Error Rate on the CIFAR-10 validation set

paper are superior in quality, are parallelizable, and require significantly less time to train. The paper's results show that the Transformer generalizes well to other tasks as it applies it successfully to English constituency parsing both with large and limited training data.

In this study of Transformers for text analysis, the objective function: the cross entropy loss function is optimized with an Adam optimizer.

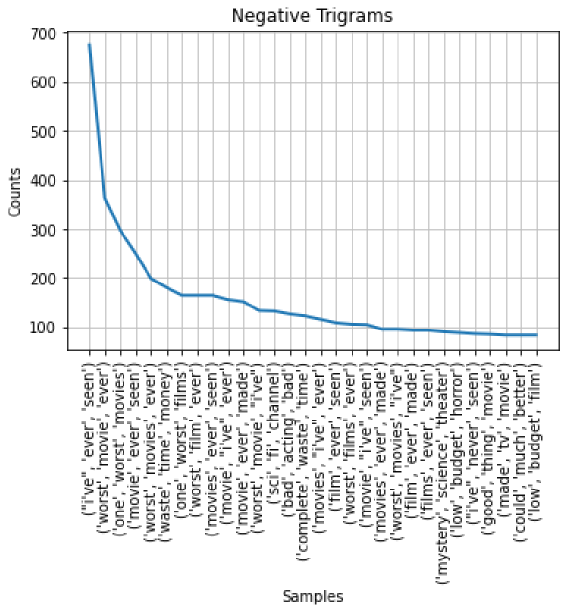
I began my study by conducting a comprehensive statistical analysis on the Reuters Newswire data set from the tensor flow keras data set. I attested my statistical analysis with data visualizations so I could easily comprehend my findings. I then trained a text classifier with a Transformer for the Reuters Newswire data set.

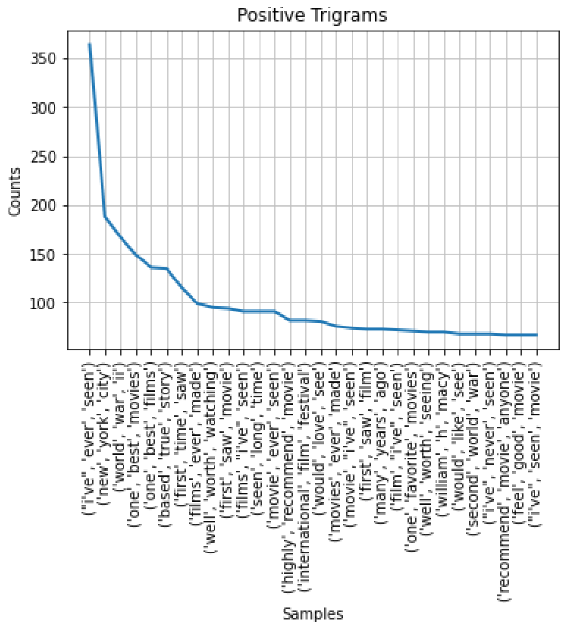
2.2 Exploratory Data Analysis

To begin with, I critically evaluated the Reuters Newswire data set. In this data set there are 8,892 training samples and 2,246 testing samples. I began my analyses by creating a histogram of Newswire lengths and plotted their frequency along the Y-axis. I also set the number of sampled words to 25,000, giving a ranking for the most frequently used 25,000 words. In my results, I found that of the 25,000 sampled words, 24,998 were unique words. From this illustration, I was able to find that most of the Newswire lengths are approximately between 1 and 250, with the average length being 145.96.

In my second analyses, I constructed a histogram that displays the top 10,000 words based upon their word lengths. My findings reveal that words with a length of 6 and 7 are the most commonly used words, with an average word length of 7.04. Next, I created a a histogram of the 30 most commonly used words. Not surprisingly, stopwords such as: the, in, and has were the most commonly used words in the data set. Although I felt that it was important to illustrate the commonality of these stopwords words, for research purposes, I decided to remove these stopwords when evaluating the text and their associative sentiment.

For my next analyses, I created three histograms of the most commonly used trigrams based upon their sentiment (positive or negative). When analyzing my results many of the most frequent trigrams appear in both positive and negative sentiment reviews. This is because some of these words are prominent throughout movie reviews in general.





2.3 Classify Text

To classify the text, I created two classes. The first class implements the Transformer. This Transformer is a multi-head attention Transformer implemented using keras.layers. Using a ReLU activation, I then stacked the Transformers using keras.sequential. I then established a residual connection for each sublayer, assigned a dropout rate, and normalized each layer. The second class implements embedded layers before the first decoder and encoder layer, this class is also responsible for mapping words to vectors. I used the Reuters Newswire data set to create the training and validation data sets. As previously mentioned above, I used a cross entropy function and an Adam optimizer to train the model.

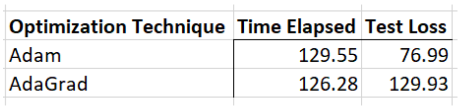
3 Part III: Variational Autoencoder for Image Analysis

3.1 Context

Part III of this project was based off the paper *β-VAE: Learning Basic Visual Concepts With a Constrained Variational Framework*. The paper presents a new framework for automated discovery of interpretable factorised latent representations from raw image data in a completely unsupervised manner by modifying a variational autoencoder framework. The authors: Higgins, Matthey, Burgess, Glorot, Botvinick, Mohamed, and Lerchner introduce an adjustable hyperparameter β that balances latent channel capacities and independent constraints with reconstruction accuracy. In this project, we explore their findings.

3.2 Optimization Techniques

In machine learning, the purpose of optimizers is to reduce the loss by updating weights, learning rates, biases, and to improve the model’s performance. In my research, I utilized two optimization techniques: Adaptive Moment Estimation (Adam) and Adaptive Gradient Algorithm (AdaGrad). AdaGrad is a stochastic optimization method that adapts the learning rate to the parameters. It performs smaller updates for parameters associated with frequently occurring features, and larger updates for parameters associated with infrequently occurring data, this is why this optimization technique is well-suited for sparse data. One of the advantages of using AdaGrad is that it eliminates the need to manually tune the learning rate. However, the problem with AdaGrad is it accumulates squared gradients in the denominator; this is a problem since every added term is positive, causing the accumulated sum to grow during training, yielding in a learning rate that eventually becomes so small that the algorithm is not able to acquire additional knowledge. On the other hand, Adam is a combination of both momentum and RMSprop; the momentum reduces noise and adds smoothing effects while RMSprop updates the learning rate in a way that is free from the disadvantages of AdaGrad. In our results, the Adam optimizer significantly out performed the AdaGrad optimizer.

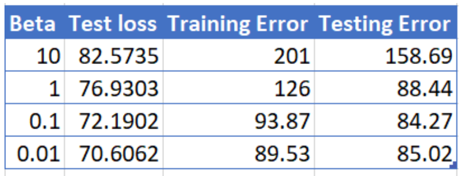


Optimization Technique	Time Elapsed	Test Loss
Adam	129.55	76.99
AdaGrad	126.28	129.93

Figure 7. Comparing Adam and AdaGrad

3.3 Regularization

When the β parameter is increased, it constrains the stochastic layer to be closer to the normal distribution; therefore, making the model less flexible. Theoretically, what we expect is for: the test loss to increase when β increases and the training error to increase with β. However, when β is large, there will be less variance and less over-fitting, and this causes the test error to drop. Nevertheless, when β is too large, the bias will increase yielding in higher test and training error. The results below illustrate this regularization theory.



Beta	Test loss	Training Error	Testing Error
10	82.5735	201	158.69
1	76.9303	126	88.44
0.1	72.1902	93.87	84.27
0.01	70.6062	89.53	85.02

Figure 8. How β effects regularization

References

[1] He, K., Zhang, X., Ren, S., and Sun, J. (2016). *Deep Residual Learning for Image Recognition.* 770-778. 10.1109/CVPR.2016.90.

[2] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). *beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework.* ICLR.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., and Polosukhin, I. (2017). *Attention Is All You Need.*