LABEL CORRUPTION AND GENERALIZATION IN DEEP LEARNING

Angela Zhuang | angelazzz.uwu@gmail.com
SHINE Lab
Arcadia High School, Class of 2024
USC Viterbi Department of Computer Science | SHINE 2023

Introduction

The usage of machine learning and neural networks has exploded in recent years, being implemented into processes such as facial recognition, search engines, and spam detection. These neural networks train on extensive quantities of data in order to produce accurate results for any situation it encounters. Thus, how accurately the training dataset is labeled affects the generalization, or how well the machine does on a test data point compared to a data point from the training set. In order for neural networks to be optimized and efficient, it is important to understand the relationship between label corruption, or inaccuracies in the labeling of data points, and generalization.

Objective & Impact

In Professor Sharan's lab, the effects of label corruption on generalization are explored. Various percentages of label corruption are applied to a dataset that the machine learning model trains on. Then, after the model finishes training, it makes predictions for the test dataset. The resulting error in the test dataset's predictions is recorded. The goal is to recreate the results of the research paper, "Understanding Deep Learning Requires Rethinking Generalization" [1], to understand how mislabeled data affects a model's performance.

Acknowledgements

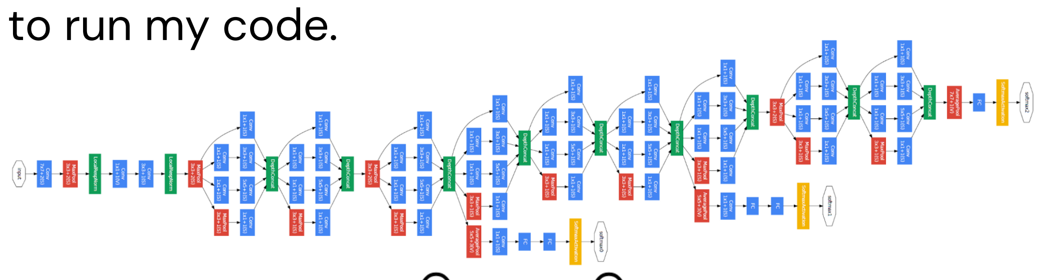
I would like to thank Professor Vatsal Sharan for this opportunity to learn more about machine learning, specifically in label corruption and generalization. Furthermore, I would like to thank my mentor, Julian Asilis, for always being there for guidance--from explaining back propagation on day one to assistance every step of the way. Additionally, I would like to thank my center mentor, Minsun Shim for all of her helpful insights and check-ins.

Research & Learning Process

The dataset I used, identical to the research paper we are recreating, is the CIFAR-10 dataset, which has 60,000 colored images across 10 categories [2].

Fig 1. Sample images from the CIFAR-10 dataset.

In order to train the models on various levels of corrupted data, I had to implement the models as well as functions to "corrupt" the data and train the models in code.
With that in mind, I referenced code from a public Github repository to write the training and data corruption functions [3].
I moved on to **implementing different models** into the code: InceptionNet, AlexNet, and Multilayer Perceptron (MLP for short). I learned to use the various libraries in **Pytorch**, such as the Pytorch.models package to import AlexNet into the program, as well as **Jupyter Notebook** to run my code.



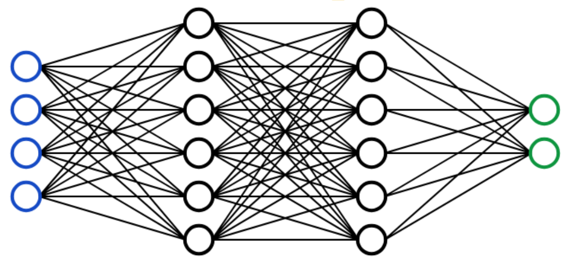


Fig 2. (Top) The InceptionNet model architecture.
Fig 3. (Bottom) An example of MLP architecture.

After the various models were imported, in the training function, I **added the hyperparameters** for momentum, where the model updates its parameters based on its prior parameter changes, and exponential learning rate decay, where the learning rate--how much the model changes its parameters--is decreased exponentially. The parameters were set to 0.9 and 0.95 respectively, as the research paper had done.

Methods & Results

1. The three models: InceptionNet of batch size 32, InceptionNet with batch size 64, and MLP, were trained on the CIFAR-10 dataset with 0% label corruption for 250 epochs (InceptionNet) or 200 epochs (MLP).

2. Utilizing **WandB**, a platform for tracking machine learning progress, the model's training and testing errors were recorded (i.e. the percentage of misclassifications on the training and testing sets).

3. Steps 1 and 2 were repeated for label corruptions from 10% to 100%, in increments of 10%.

4. Using the results from WandB, Matplotlib was utilized to plot graphs of label corruption's effects on the final test and train error of the model. The "Label Corruption on Test Error" graph is a recreation of Figure 1c in the research paper [1].

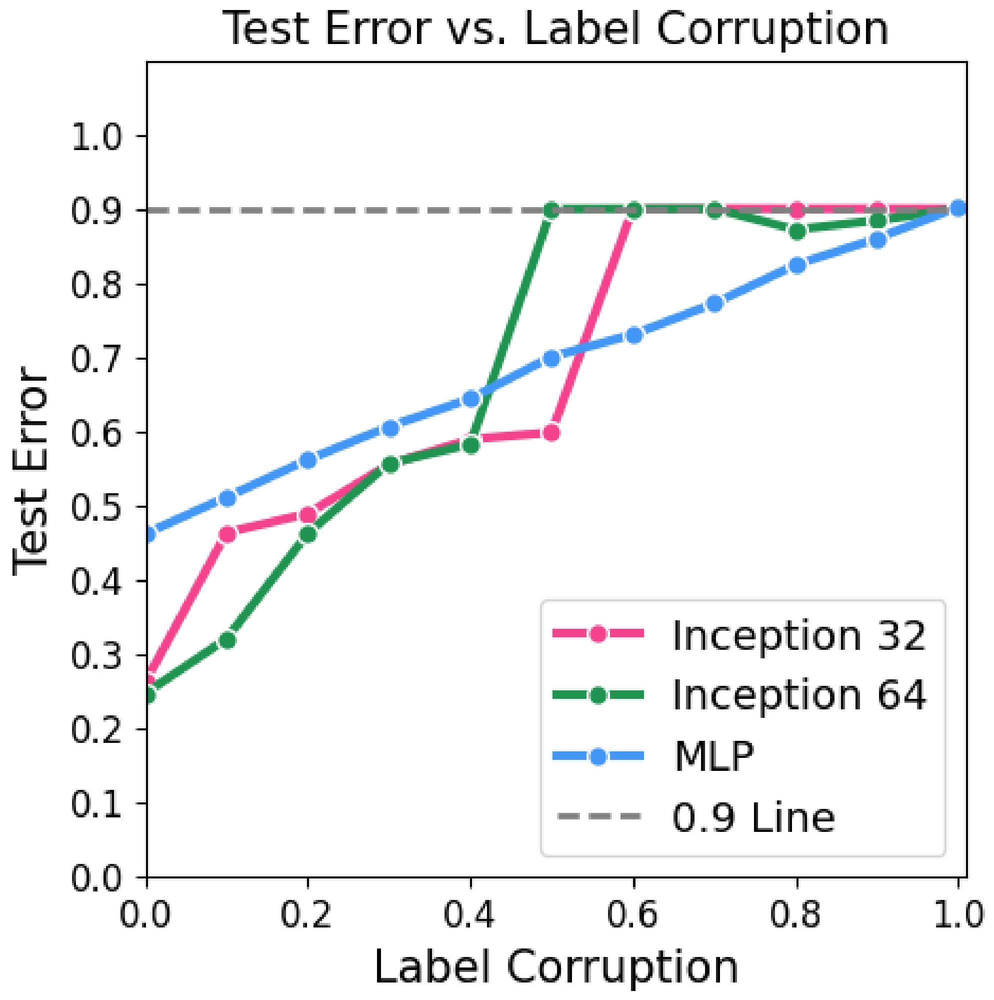


Fig 4. Label Corruption on Test Error. The final test errors for each level of label corruption.

Results Analysis

At 0% label corruption, each machine learning model is able to train very well, with their train errors being nearly 0% at the end of their training. Additionally, they perform acceptably well on the test dataset, with test errors of about 25% for InceptionNet and 45% for MLP. This is reasonable given MLP's simplicity. As the label corruption increased, the test error of the models increased as well, indicating a positive correlation. The MLP model was able to get to very low training errors even with high label corruption, e.g. having nearly 0% train error with 100% label corruption. Of course, it suffers 90% test error on fully corrupted training data. This demonstrates how neural networks can do exceptionally well on training sets even if they are completely nonsense, and thus generalize poorly. Therefore, it successfully illustrates how a machine learning model's generalization is not due to intrinsic properties of model architecture, and label corruption should be avoided to train an optimal machine learning model.

Next Steps

In order to more accurately recreate the results of the research paper, InceptionNet should be trained to 0% train error, then analyzed accordingly. We struggled to reproduce this aspect of the paper, as did several others online. Moreover, I wish to use other neural networks, such as AlexNet, and further see the effects of label corruption on generalization.

Citations

[1] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, arXiv, https://arxiv.org/abs/1611.03530 (accessed 2023).
[2] A. Krizhevsky, "The CIFAR-10 Dataset," CIFAR-10 and CIFAR-100 datasets, https://www.cs.toronto.edu/~kriz/cifar.html (accessed Jul. 19, 2023).
[3] mdv3101, "MDV3101/rethinking_generalization: SMAI project: Understanding deep learning requires rethinking generalization," GitHub, https://github.com/mdv3101/Rethinking_Generalization (accessed Jul. 19, 2023).