ICLR 2019 Paper Reproduction - An Empirical Study of Example Forgetting During Deep Neural Network Learning

1 Introduction

In this report, we discuss our replication efforts on the paper with title "An Empirical Study of Example Forgetting During Deep Neural Network Learning." We replicated the following experiments from the paper. We classified forgettable/unforgettable examples for the MNIST, permutedMNIST and the CIFAR-10 datasets. We also visualized the top 10 forgettable vs unforgettable examples. Then we verified the performance of a model with varying percentage of ordered unforgettable examples removed. Finally for the CIFAR-10 we compared the forgettable examples experiment by running on more than one architecture. We implemented the convolution neural nets as described in the paper from scratch. However, we used the existing implementation of Res-net with cutout from the soure linked in the paper, `https://github.com/uoguelph-mlrg/Cutout`.

2 Forgettable Examples

Our graphs for the MNIST, permutedMNIST match the graphs shown on the paper. We also seemed to get around the same actual count of events of forgettable and unforgettable examples. We trained our model on 2 different seeds to ensure that simply the ordering did not affect our results. This was also done in the original paper. We computed the correlation between the 2 seeds which was a correlation of 0.8559372. This is simply the correlation between the forgetting events per example as computed between the 2 models. Furthermore, the 95% confidence interval was 0.8535760 to 0.8582634. This result indicates that the particular order of the data did not affect our results. In other words, the forgetting events per example are simply not due to chance or the specific order in which the training data occurred in the dataset.

However, for CIFAR-10 our graph did not fully match the one provided in the paper. We had a smaller percentage of examples with 0-1 forgetting events as opposed to the original which has close to 0.3 percent while we only have 0.2 percent. The difference probably arises from the fact that our Res-Net was configured from slightly different setting than the one specified in the paper.

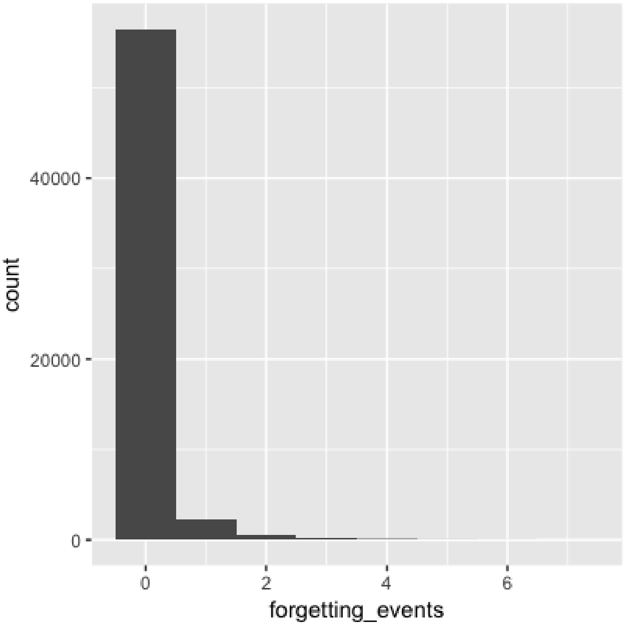


Figure 1: MNIST Forgettable Examples

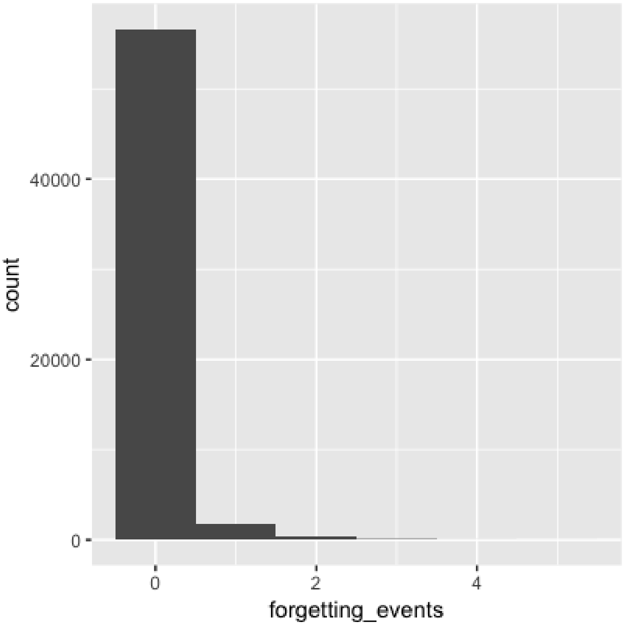
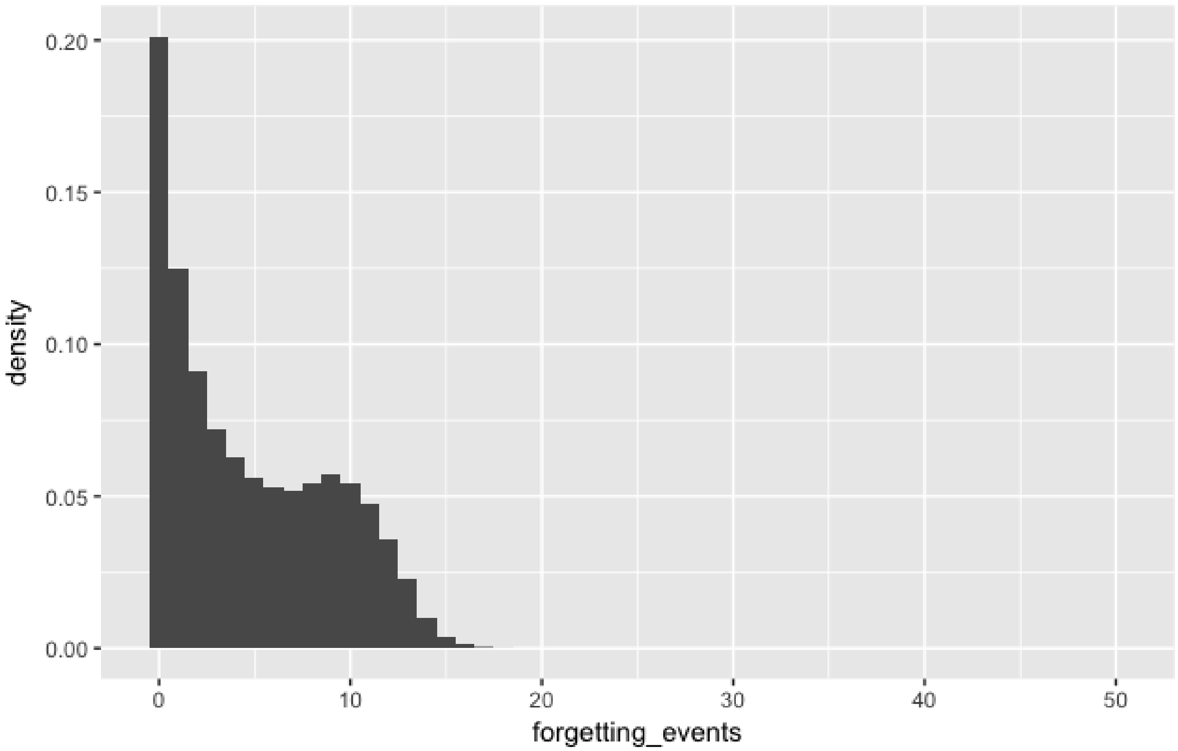


Figure 2: PermutedMNIST Forgettable Examples

It could also be different because we trained for fewer epochs than the paper which was due to our time and resource limitations. We trained for about 50 epochs while the original authors have trained the model for 200 epochs.



3 Visualizing Forgettable & Unforgettable Examples

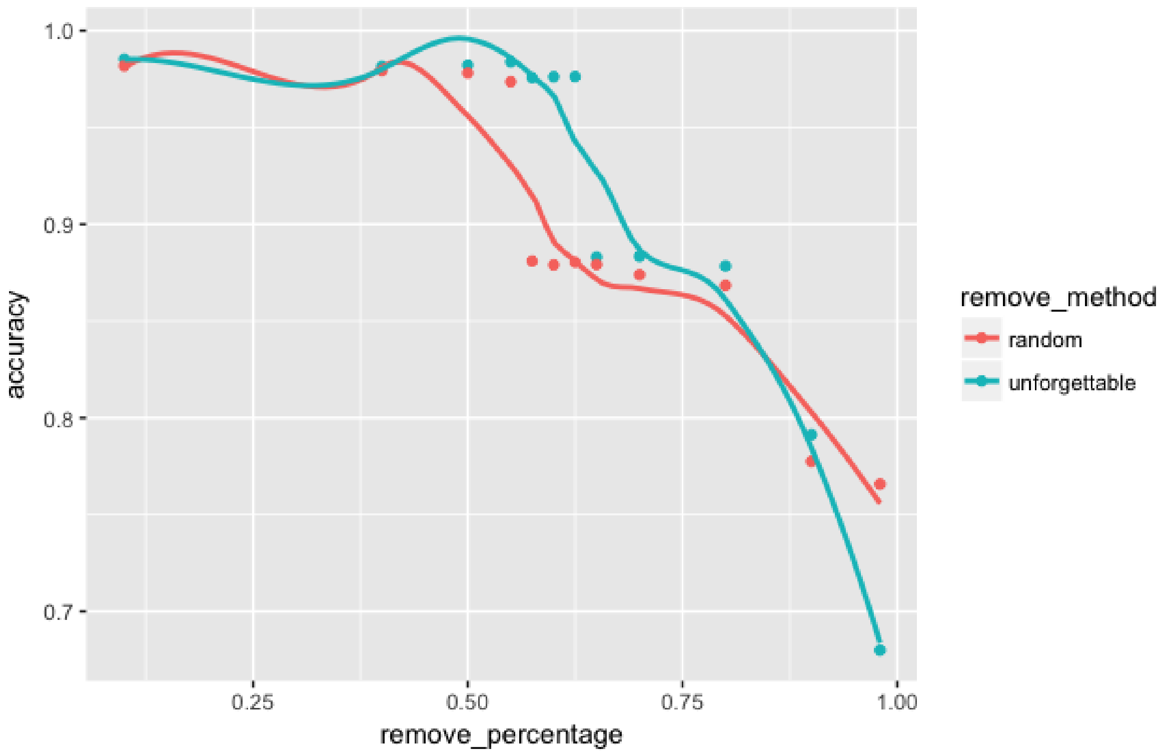
We visualize the top 10 forgettable vs unforgettable examples. This was also done in the original paper. Similar to the paper, we also notice why the unforgettable examples are unforgettable. They seem to have clear images on a plain and dissimilar background as opposed to the foreground. Many of the top forgettable ones seem to blurry, unclear images.

Top 10 unforgettable Examples

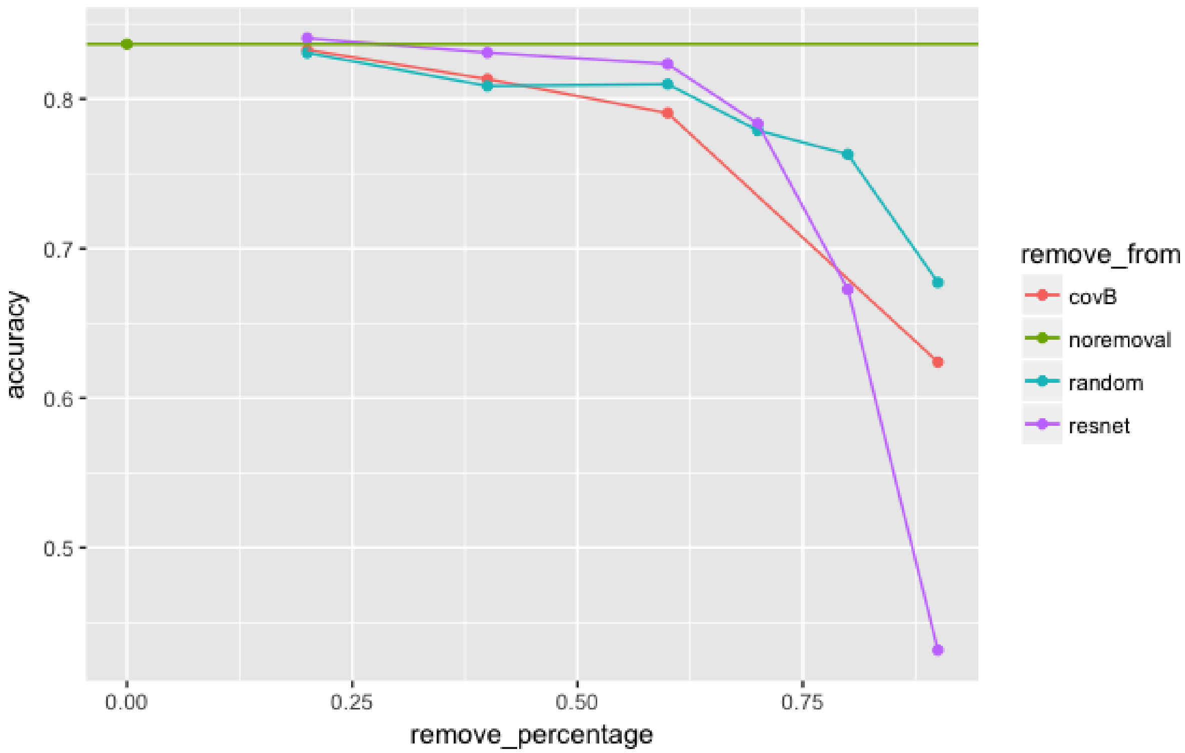
Top 10 Forgettable Examples

4 Removing Unforgettable Examples

We had interesting results when we removed unforgettable examples from MNIST. As claimed by the paper, removing up to 30 percent of the examples in the order of forgetting events (least to greatest), the training accuracy is not really affected. At 50% we can see a dramatic difference between the selectively removed and the randomly removed as the paper claimed. However, after 50 percent of the points are removed, the selectively removed examples decrease at a faster rate and goes below the randomly removed example accuracy. However, we are still confident that removing up to 30% of the data in order of forgetting events (least to greatest) does not hurt test accuracy. This is a very promising result for future work in the field to figure out ways to speed training of models by identifying certain examples before hand and removing them from training to speed up the process.



Given that it worked for MNIST, we also tested it for CIFAR-10 dataset. We ran two different architectures for training on the CIFAR-10 dataset. The results are shown below.



From the result above we can see that once again that removing up to 30% of the data in order of forgetting events does not affect accuracy much at least for the Res-Net which was the most powerful architecture used. A surprising result which is also observed when removing from the MNIST dataset is that when removing a higher percentage of examples, the selectively removed ones drop faster than the random removal. The original paper did not remove beyond 40% of the data so the result of removing more is unknown.

Why might the RES-NET trained forgetting event ordering removal drop faster than random removal when more than 70% of the training examples are removed? Our guess is that the training algorithm somewhat resembles the way we learn from solving problems. When we order the example in terms of forgetting events from least to greatest, it can be thought of as easy to hard examples to learn. The training performance is not affected when 30% of the "easy" examples are removed but when you remove a large percentage of "easy" examples, it hurts the learning of the algorithm. When we try to learn say practice for coding interviews, skipping easy examples and working more towards the harder example would probably save us time and help us achieve our goal of getting better at the problems. However, skipping too many easy examples would probably hurt our performance because it is now harder to learn hard examples. It would make sense to work your way up to harder examples.

This is our guess as to why the accuracy drops faster for selective removal as opposed to random when the removal percentage is significantly high.