Outlier Removal for Deep Neural Networks

Sanjeet N. Dasharath

Australian National University, Canberra ACT, Australia
u6138440@anu.edu.au

Abstract. In this paper I try to apply the technique of Bimodal Distribution Removal [1] to a deep convolutional neural network on a subset of the CIFAR10 [2] dataset. I found that the results of applying the BDR technique to a deep learning problem improved test accuracy on the dataset even though the technique from the original paper does not directly translate to large datasets. I discuss a few strengths of the BDR algorithm, why it may have improved the test set accuracy, as well as provide some reasons for why it may not be suited for deep learning applications.

Keywords: Deep learning · Machine learning · Pruning.

1 Introduction

The application of 'deep' neural networks has found immense recent success in several areas including computer vision, natural language processing, etc. This advent of deep learning has mainly been propelled by the availability of computational power to apply these techniques to very large datasets.

As the demand for testing deep learning related solutions for new problem domains increases, so does the need for gathering large datasets related to these domains. However, gathering extremely large labelled datasets is a difficult and tedious task. Labelling generally needs to be done manually and this can introduce several noisy, or mislabelled, samples into the dataset. Thus, there is a need for the network to automatically be able to detect noisy data samples and prune the dataset dynamically, while training, to weed out these samples.

A highly successful technique for pruning the training set from noisy samples was the Bimodal Distribution Removal (BDR) algorithm introduced by Slade & Gedeon [1]. However, this technique is mainly applied to problem domains with small datasets and simple MLP networks. In this paper, I first describe the BDR algorithm in detail and then explain the results I got for applying it on a deep convolutional neural network and the CIFAR10 [2] dataset.

2 Bimodal Distribution Removal

The BDR technique is a method for automatically detecting outliers during training. The main idea is that initially the distribution of error frequencies on the training set is spread out. Thus its variance is high. However, after training for a while, the distribution becomes roughly bimodal. The lower error peak represents examples which were successfully trained. The higher error peak includes slow learning examples, as well as outliers.

Once the variance becomes sufficiently low, we calculate the mean error $\overline{\delta}_{ts}$. The key assumption here is that the mean will be heavily skewed towards the low error peak. Thus, the first step in isolating the outliers is to consider all those samples with error greater than $\overline{\delta}_{ts}$.

Next we calculate the mean $\overline{\delta}_{ts}$, and with it the standard deviation σ_{ss} of this subset. Finally, we remove all training examples from the original training set for which

$$error \geq \overline{\delta}_{ts} + \alpha\sigma_{ss}$$

where $0 \leq \alpha \leq 1$.
This process is repeated until the variance attains a certain value.

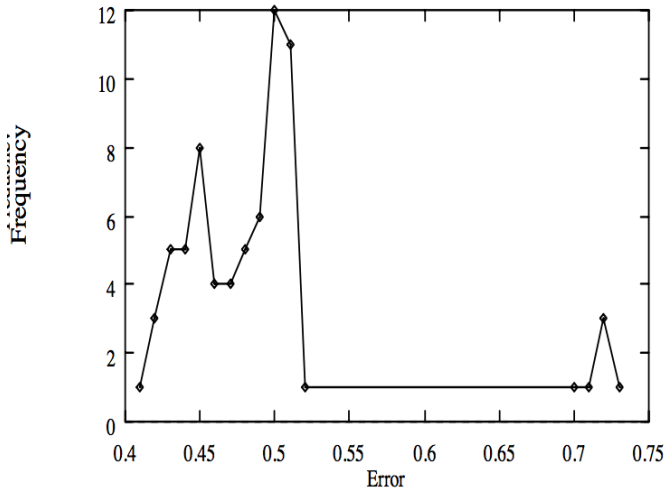

Fig. 1. Error distribution at epoch 0

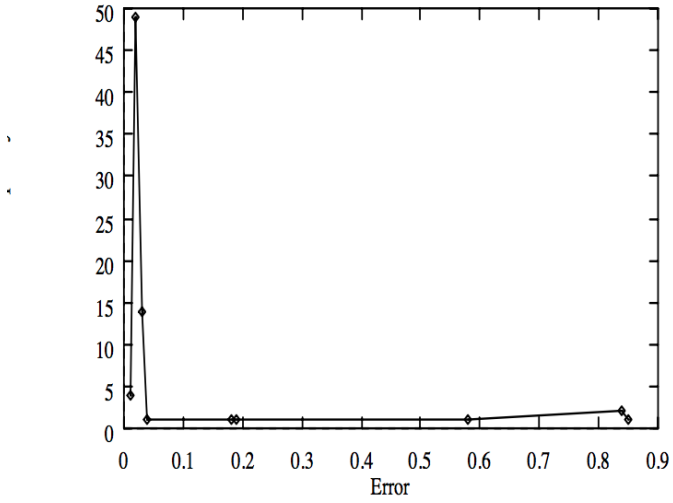

Fig. 2. Error distribution at epoch 500

3 Implementation

The architecture I use for the classification task is a deep neural network with two Convolution [3] layers followed by two fully connected layers. The convolution layers use no pooling, and the LeakyReLU nonlinearity. Between the first and the second hidden layers, I also use a Dropout [4] layer as a regularizer.

The network is trained using back-propagation and the Adam [5] optimizer with a learning rate of 10^{-4}. I train the network on a subset of the CIFAR10 dataset using only 20,000 examples for training as well as 20,000 for testing. Instead of minibatches, the network is trained on the entire dataset at once. This makes it a lot easier to implement pruning of the dataset while training. Finally, the network is trained for 20 epochs.
Unlike the original BDR paper, I begin pruning the dataset every even epoch starting from epoch 12 onward. This is because unlike in the original paper, the error variance on this problem starts off lower than 0.1. Thus, I instead treat the start and stopping conditions of the pruning step as hyperparameters. I found starting at epoch 12 with pruning happening on even epochs as optimal using a standard search procedure for hyperparameter testing.

4 Results and Comparison

After training the network for 20 epochs with, and without BDR, I found the following results:

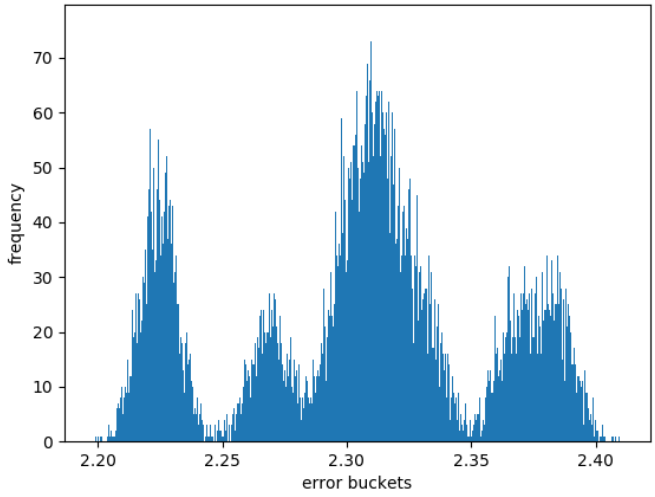

Fig. 3. Error distribution at the start of training

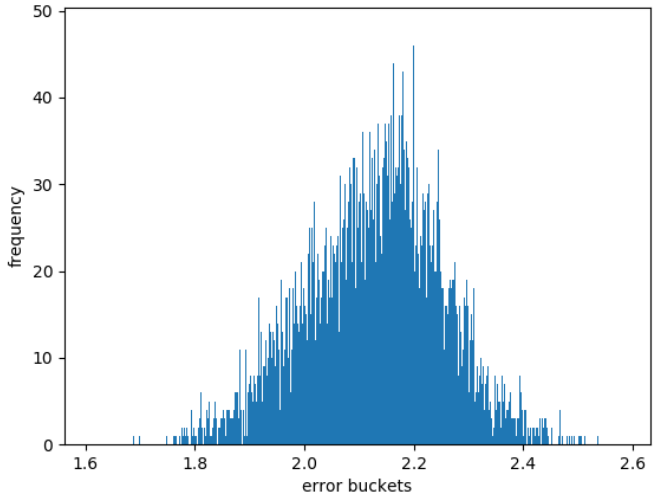

Fig. 4. Error distribution after BDR

Thus, using the BDR technique, I get an improved test accuracy using the same model.

The current state of the art performance on CIFAR10 is an accuracy of 96.53% on the test set [6]. These type of performance results require training on multiple GPUs for several days or weeks, and was infeasible for my experiment. Thus comparisons with state of the art results is at present not possible, however is a suggested future work.

5 Conclusion

One of the reasons why BDR gives improved accuracy is likely because deep neural networks are very complex. Thus the removal of noisy or hard to train data from the training set may be helping the network focus on learning the "good" samples while not having performance dragged down by the outliers.

While this may seem promising, I believe BDR may also cause the network to overfit. Since BDR prunes the training set, it continually makes it easier for the network to train on the remaining data. This has the potential to

Table 1. Results

Technique	Test Accuracy
None	8.19%
BDR	17.19%

to get the model to severely overfit.

Finally with large and complex datasets like CIFAR10 with input dimensionality in the thousands, the variance criterion of the original paper does not translate well and we are forced to come up with other heuristic measures for when to begin and stop pruning. This can be a time consuming process, and may need lots of trial and error.

Being able to detect and remove outliers will remain an important task in machine learning. The success of deep learning is in part due to the massive sizes of datasets used. These generally reduce the impact of outliers on performance however exploration of solutions to this problem will still prove valuable.

6 Future Work

The technique of BDR as presented in this paper uses the entire training set. A possible extension would be to find a similar technique which works on minibatches. This could then become a practical technique for modern deep learning problems which almost exclusively use minibatches.

An important suggestion for future work would be to train the network on multiple GPUs and perhaps using other datasets. Since this was not possible for my experiments, test accuracy was very low.

Finally, the last suggestion for more work would be to find an extention of the BDR algorithm for recurrent neural networks. As it is right now, it perhaps is not suited for many problems in domains where sequential data are used.

References

1. Slade, P., & Gedeon, T. D. (1993, June). Bimodal distribution removal. In International Workshop on Artificial Neural Networks (pp. 249-254). Springer, Berlin, Heidelberg
2. Krizhevsky, Alex. (2012). Learning Multiple Layers of Features from Tiny Images. University of Toronto.
3. Lecun, Yann & Bottou, Leon & Bengio, Y & Haffner, Patrick. (1998). Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE. 86. 2278 - 2324. 10.1109/5.726791.
4. Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res., 15(1):19291958, January 2014.
5. Kingma, Diederik P. and Ba, Jimmy. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG], December 2014.
6. Graham, B.: Fractional max-pooling (2014). arXiv:1412.6071