A Comparison Between Bimodal Distribution Removal for a CNN versus a NN

Joshua Hull

Research School of Computer Science
The Australian National University
u5801055@anu.edu.au

Abstract. Noise is often added to training data to push the network to generalise better. But the impact of real-world noise on training data is not easy to replicate. Therefore, it is natural to seek for ways to handle real-world noise directly.
I have taken a method of noise reduction which has seen success at improving the generalisation of neural networks, and shown that it's effect on CNNs is promising, and it's improvement to run-times is significant.
Keywords: convolutional neural network, bimodal distribution removal

Introduction

Convolutional Neural Networks (CNN) are a type of Deep Neural Network often used to process grid-like data such as images. A CNN identifies features within the input data through the use of filters. Each filter is able to identify some feature of the data. There may be a filter to detect eyes or a nose, for example. A CNN will pool the activation of these filters together and identify some greater feature such as a face.

CNNs can struggle handling noisy input data. Going back to the example of an image of a face: noise may come in the form of glasses which most faces in the set do not have, or there may be noise because the face is obscured. The former case is an acceptable but uncommon identification of a face, whilst the latter may be better identified as something else – the hand waving in front for instance.

We will take a successful method of noise reduction for "simple" (non-deep) Neural Networks devised by P. Slade and T.D. Gedon **[1]**, and apply it to a CNN. This report will compare the effectiveness of that method – Bimodal Distribution Removal – on a simple NN versus a CNN.

The data I have chosen to evaluate my CNN is the CIFAR-10 data-set. It consists of 10 classes of images, and 6,000 images for each class. The classes are: plane, car, bird, cat, deer, dog, frog, horse, ship, truck. Each image is 32×32 pixels in dimension.
I have chosen this data-set because many people before me have researched and evaluated their CNNs upon this data. It is also readily available, and neatly formatted.

Bimodal Distribution Removal is the method of noise reduction I have applied to the CNN. Slade and Gedeon showed this method is a better choice than Outlier Detection, Absolute Criterion, and others **[1]**.

Reproducing Past Experiments

Before we can compare the effectiveness of BDR on Deep Learning against BDR on a simple Neural Network: we will reproduce the results of Slade and Gedeon, and demonstrate proper implementation of BDR.

The data I have chosen to replicate the aforementioned experiment consists of 315 instances of stock portfolio performance statistics. There are 12 features representing various risks and returns within the stock market. The final column (I have removed a few and focused on just one prediction) is the Total Risk of the investment. If the Total Risk is > 0.5, it is considered high risk, else low risk.
I have chosen this data-set since it is similar in nature (continuous features, continuous prediction) to that of Slade's and Gedeon's experiment.
The data-set is split randomly into 66% (208) training data, and 34% (107) test data.
To make predictions on the data, I am using a simple Neural Network, running for 500 epochs.

This is the result of training the Neural Net on the training data, then having it make predictions on the test data without BDR:

Trial	Accuracy
1	83%
2	84%
3	90%
4	75%
5	85%
6	86%
7	85%
8	94%
9	95%
10	91%
Average	86.8%

Fig 1. Prediction accuracy of a NN without BDR.

Without BDR, the Neural Net guesses correctly whether an investment is high or low risk ~87% of the time.

These are the results of the same Neural Net running for the same number of epochs but with BDR:

Trial	Accuracy
1	89%
2	94%
3	88%
4	88%
5	85%
6	95%
7	90%
8	93%
9	85%
10	92%
Average	89.9%

Fig 2. Prediction accuracy of a NN with BDR.

With BDR, the Neural Net guesses correctly whether an investment is high or low risk ~90% of the time.

These results align with the improvement Slade and Gedeon demonstrated **[1]**. We have successfully implemented BDR.

Method

Bimodal distribution removal occurs in these steps:

1. make a prediction for every training pattern
2. calculate the average error of that prediction across all patterns
3. calculate the standard deviation of that error
4. remove outlier patterns from the training set

For average error calculation, I am using Mean Squared Error:

$$E(X, Y) = \frac{\sum_{n=0}^{N} (Y_n - X_n)^2}{N}$$

The standard deviation is:

$$o = \sqrt{\frac{E(X, Y)}{N - 1}}$$

Where X is the list of training patterns, and Y is the list of predictions for those patterns.

Every z epochs (where z is up to you), the network removes every training pattern with $e(x, y) > E(X, Y) + o$. That is, if the error of in the prediction for any training pattern is further than 1 standard deviation from the mean, remove that training pattern from the list.

In the case of the Convolutional Neural Network, images which may have been particularly noisy or unclear were removed from the training set via this method.

Results

Each CNN ran for 4 epochs, and ran BDR every 2 epochs. The epoch count is so low since we do not necessarily need to show that a CNN can achieve accurate predictions, but that it may achieve predictions that are not random. For 10 classes, a 10% accuracy could be considered random.

For each class in the data, we can see the accuracy at which the CNN correctly guesses a particular image is that class.

These are the results over all 10,000 test images without BDR:

Class	Accuracy
Plane	30%
Car	69%
Bird	36%
Cat	50%
Deer	36%
Dog	22%
Frog	57%
Horse	58%
Ship	68%
Truck	58%

Fig 3. Prediction accuracy of a CNN without BDR.

The average accuracy across all images is 48%.

These are the results over all 10,000 test images with BDR:

Class	Accuracy
Plane	48%
Car	67%
Bird	32%
Cat	32%
Deer	37%
Dog	41%
Frog	64%
Horse	51%
Ship	61%
Truck	31%

Fig 4. Prediction accuracy of a CNN with BDR.

The average accuracy across all images is 46%.

The average accuracy is the most interesting statistic here. The accuracies of each class do not just change between BDR and non-BDR, but between consecutive executions of the same configuration of the CNN. I suggest this is because the CNN falls into local minima and will tend to, for example, guess dog rather than cat.

The difference in average accuracy is small but consistent. I have run each network a few times each and the non-BDR CNN seems to perform slightly better.

I have recorded the running time of the CNN without BDR to be ~4 minutes, 22 seconds, 94 milliseconds.
The CNN with BDR: 3 minutes, 47 seconds, 46 milliseconds.
This is a 13.5% speed increase.
It is clear from these results that BDR has the potential to significantly speed up the training time of a CNN, with the accuracy of prediction being only slightly worse than otherwise.

Discussion

Liangji Zhou, Qingwu Li, Guanying Huo, and Yan Zhou developed an extension to CNNs by combining them with Biometric Pattern Recognition **[3]**. Zhou et al. showed their method out-performed traditional CNNs and CNNs combined with Suport Vector Machines, as well as a few other methods.

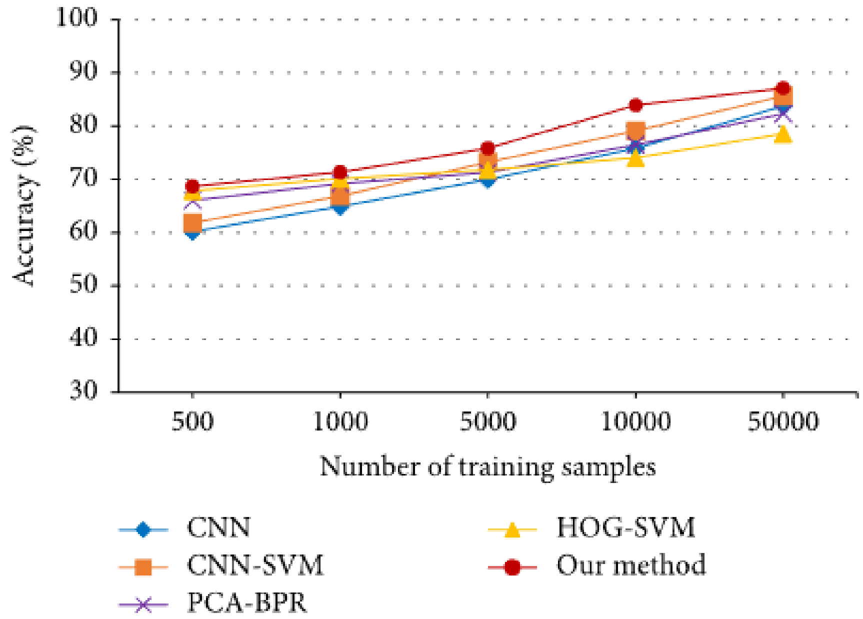

Fig. 5: Zhou et al. Performance **[3]**

Note: their accuracy is around 80% where mine was around 50% due to their goal being slightly different than mine. They show their method is able to outperform a regular CNN to a similar degree that a I have shown a regular CNN outperforms BDR. However, Zhou et al. note significant slow down of running time when combining algorithms with CNN, whereas BDR demonstrated an improvement in that regard.

Slade and Gedeon showed BDR may improve the performance of a simple neural network. They also noted that BDR should lead to decreased run-times of neural networks **[1]**. Whilst the former finding has not transferred to CNNs, the latter has.

Conclusion

BDR may be used to reduce the training time of a Convolutional Neural Network without greatly impacting the accuracy of the CNN. Slade and Gedeon showed the BDR method performs well at generalising from input data into a simple neural network. I have showed the generalisation is not greatly affected when the same method is applied to CNNs, but the run-time improvements are clear.

Future Work

Zhou et al. assessed CNNs being combined with other methods including their own. Future research may include determining if and how BDR enhances those methods.

Beyond image recognition, BDR may be applied to language processing or other Deep Learning tasks. The effect of BDR may not always be to speed up the training.

List of References

1. Slade, P., Gedeon, T.D. (1993). *Bimodal Distribution Removal*. New South Wales: The University of New South Wales, pp. 1-7.

2. (2016). Stock portfolio performance Data Set. UCI Machine Learning Repository. Available at <http://archive.ics.uci.edu/ml/datasets/Stock+portfolio+performance>

3. Liangji Zhou, Qingwu Li, Guanying Huo, and Yan Zhou. (2017). Image Classification Using Biomimetic Pattern Recognition with Convolutional Neural Network Features. Computational Intelligence and Neuroscience, vol. 2017, Article ID 3792805, pp. 1-12. Available at: <https://www.hindawi.com/journals/cin/2017/3792805/cta/>