COMP 551 - Applied Machine Learning
Mini-project 4

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

In this project, we investigated the effects of dropout on different vision tasks. We replicated the paper "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", by Srivastava, N et al (1). This paper discusses dropout, a technique used in large neural networks to prevent overfitting. Dropout drops randomly selected units (and their connections) during training, creating many different thinned networks that are combined to approximate the effect of averaging their predictions. This approach leads to improved performance on various supervised learning tasks, such as vision, speech recognition, document classification, and computational biology, and achieves state-of-the-art results on benchmark datasets. The limited availability of the code used for this study made the group implement the code and then as the second phase, work on the model hyperparameter tuning. The results obtained were similar to those obtained for the study. The study was conducted around the significance of including dropout in the model to prevent scenarios of overfitting. We experimented with the datasets like MNIST, and Street View House Numbers data set (SVHN).

Submitted to ML Reproducibility Challenge 2020. Do not distribute.

1 Introduction

"Dropout: A Simple Way to Prevent Neural Networks from Overfitting", by Srivastava, N et al (1), presents a dropout method and shows that it can significantly improve the performance of neural networks on various tasks. This article showed that dropout can be used with different kinds of neural networks and different kinds of regularization techniques. The paper implemented dropout in a more complex neural network architecture and tested it on several benchmark datasets such as MNIST, CIFAR-10, and CIFAR-100 than the work that we have come up with. Regardless, the results showed that dropout consistently improved neural network performance reduced overfitting, and improved generalization on these tasks.

The technique randomly omits some neurons in the neural network during training to prevent overfitting. This technique forces the remaining neurons to learn more robust features and avoids relying too much on individual input functions. The paper provided the rationale for the dropout method and showed that it improves the performance of neural networks on various benchmark data sets. We also show that the dropout method is more effective in preventing overfitting than other regularization methods. Additionally, the paper explained the implications of hyperparameters such as dropout probability and provided guidance for choosing appropriate values. The study also performed an ablation study to demonstrate the importance of the dropout mechanism and its impact on network performance.

2 Scope of reproducibility

This article first introduces the problem of overfitting in deep neural networks and the need for regularization techniques to address this problem. We then propose dropout as a regularization technique that randomly omits (i.e., zeroes out) some of the units of the neural network during training. This prevents units from adapting together and forces them to learn more robust features that are useful for multiple class inputs.

This paper shows that dropout significantly improves the generalization performance of neural networks on various benchmark datasets such as MNIST and CIFAR-10. It also depicts the dropout outperforms other regularization techniques such as L1 and L2 regularizations and that even better performance can be obtained in combination with these techniques. In this paper, they further explore the theoretical underpinnings of dropout and propose a form of model averaging in which the various network architectures obtained by unit dropout can be viewed as a set of exponentially many sparse models. It shows that it can be considered to have connections to other techniques such as bagging and committee machines.

Overall, this paper argues that dropout is a simple but effective regularization technique that can significantly improve the generalization performance of deep neural networks across different datasets. We provide evidence to support this claim, showing that dropout outperforms other regularization techniques in terms of performance and computational efficiency based on the models that we designed for MNIST, and SVHN.

3 Datasets

The paper analyzed dropout results for 7 datasets: MNIST, SVHN, CIFAR-10, ImageNet, TIMIT, Reuters, RCV1, and Alternative Splicing. We decided to only do experiments on two of the vision subsets of these datasets: MNIST and SVHN. These data sets include different image types and training set sizes. Models which achieve state-of-the-art results on all of these data sets use dropout. The MNIST data set consists of 28×28 pixel handwritten digit images. The task is to classify the images into 10 digit classes. The Street View House Numbers (SVHN) Data Set (Netzer et al., 2011) consists of color images of house numbers collected by Google Street View. The part of the data set that they used in their experiments consists of 32×32 color images roughly centered on a digit in a house number.

4 Methodology

As the code for the study was not available, we developed the models by ourselves after referring online. The original study analyzed the impact of including dropout in the neural networks model. The models were implemented like a simple neural network model with a hidden layer. For testing the claims of the study, we were able to come up with two instances where models included dropouts and without dropouts. The hyperparameters that were considered for the verification include learning rate, batch size, number of epochs and dropout rate.

Initially, the model is trained on each of the considered datasets and evaluates its performance using accuracy. It then repeats the training and evaluation process with the same model architecture but without dropout regularization. Additionally based on the performance, it was clear that the number of epochs did have a significant impact on the accuracy and the performance was seen to improve with the number of epochs. Most of our experiments were conducted on Google Colab and jupyter notebook. Overall, Dropout has been shown to be an effective and widely used regularization technique, but there is still room for improvement and further research on its effectiveness in different tasks and domains, of which a few will be discussed in the upcoming sections of this report.

4.1 Hyperparameters

Describe how the hyperparameter values were set. If there was a hyperparameter search done, be sure to include the range of hyperparameters searched over, the method used to search (e.g. manual search, random search, Bayesian optimization, etc.), and the best hyperparameters found. Include the number of total experiments (e.g. hyperparameter trials). You can also include all results from that search (not just the best-found results).

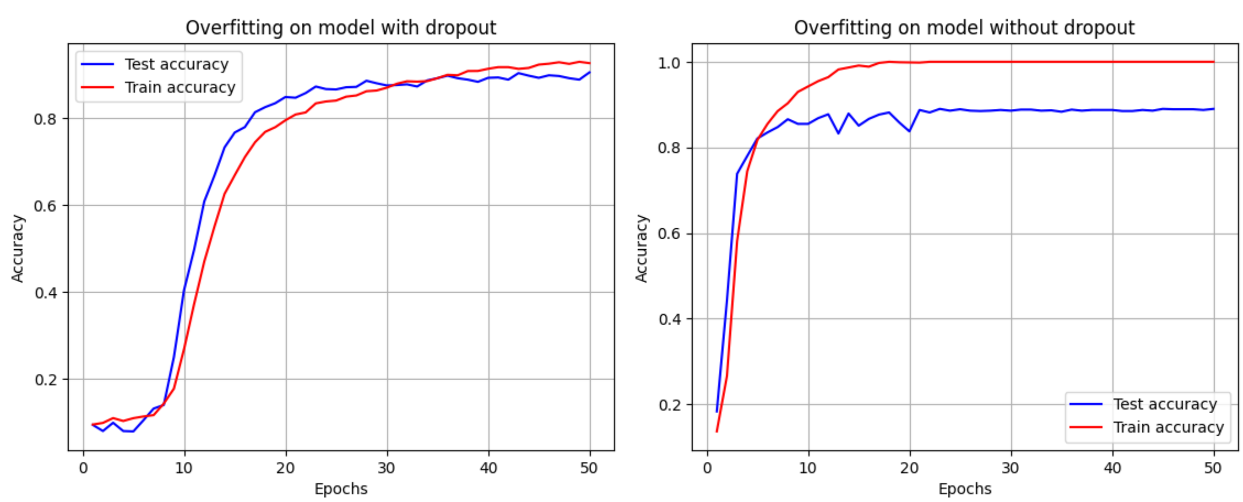
5 Results

Overall, we saw improved accuracy on both datasets when using dropout. This is in line with findings in the paper. For both models, accuracy was quite close between sets using dropout and not using dropout. However the paper, on average, actually reported smaller differences between dropout and non-dropout models. This could be due to differences in models/hyperparameters, though their original code was not supplied.

5.1 Results reproducing original paper

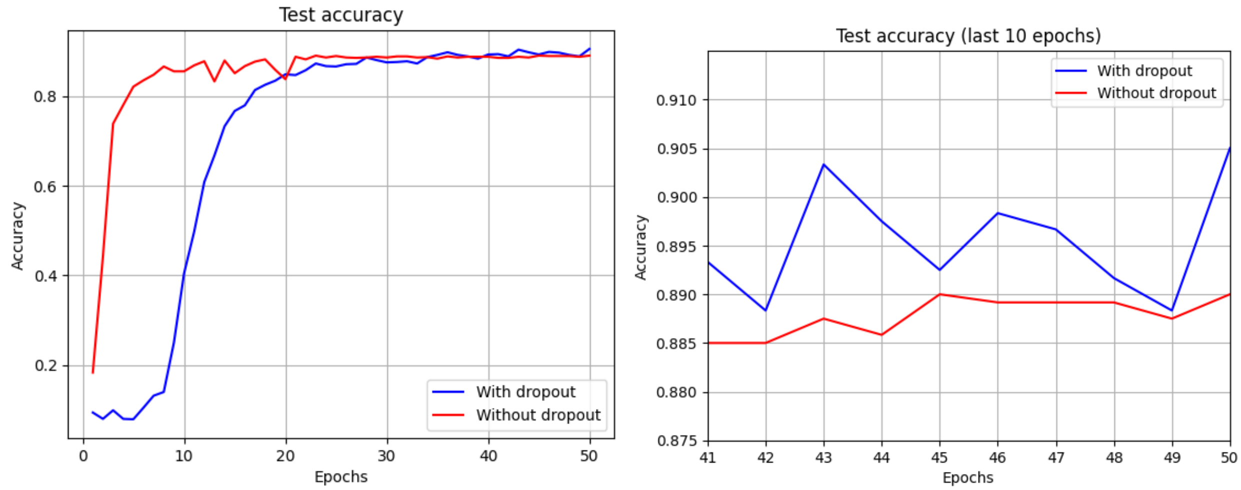
5.1.1 Result 1 - SVHN Dataset

As seen in the figures, the SVHN model with dropout had an improved accuracy of 0.905 as compared to 0.89 without (for testing). This is in line with the paper's results, which also demonstrated improved testing accuracy with dropout. We found 2% difference between dropout and non-dropout versions, while the paper reported a 1.4% difference.



(a) SVHN - Model Performance with Dropout (b) SVHN - Model Performance without Dropout

Figure 1: SVHN Dropout Effects



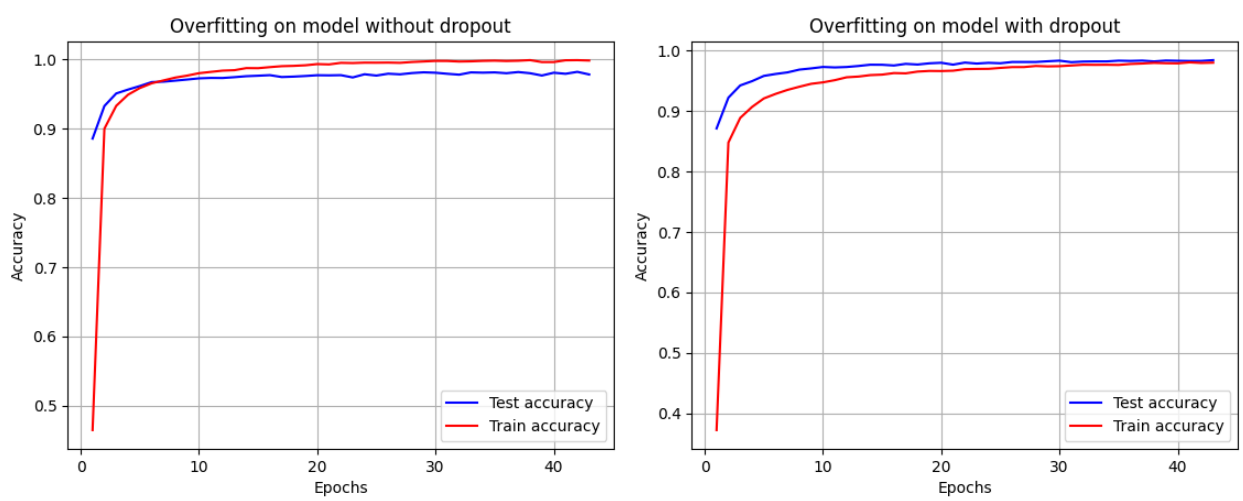
(a) SVHN - Test Accuracy

(b) Final epochs For Accuracy

Figure 2: SVHN Accuracy

5.1.2 Result 2 - MNIST Dataset

As seen in the figures, the MNIST model with dropout had an improved accuracy of 0.988 as compared to 0.979 without (for testing). This is in line with the paper's results, which also demonstrated improved testing accuracy with dropout.



(a) MNIST - Model Performance without Dropout

(b) MNIST - Model Performance without Dropout

Figure 3: MNIST Dropout Effects

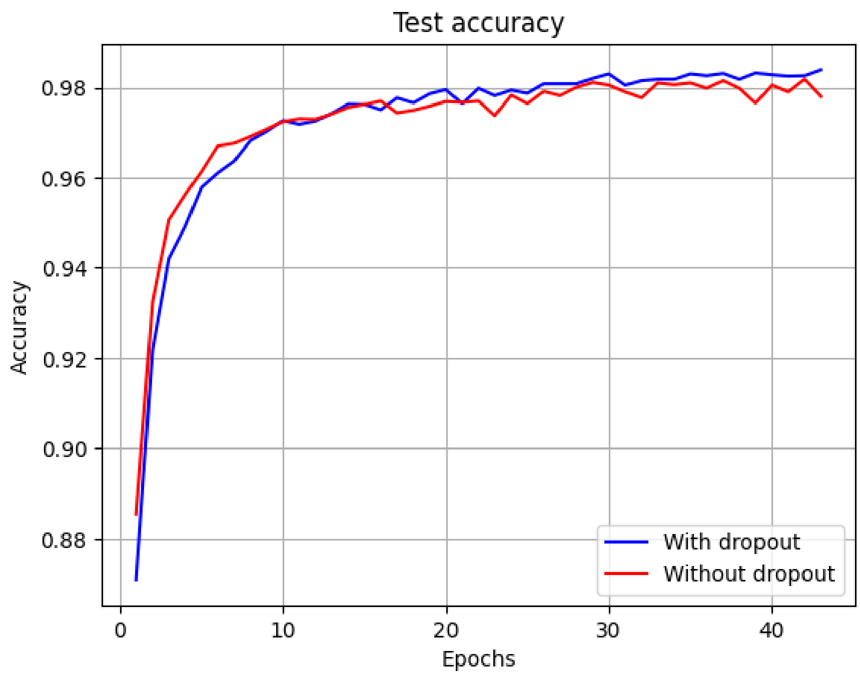


Figure 4: MNIST - Test Accuracy

6 What was easy

Overall, we found online libraries (such and tensorflow and keras) that made implementation of the models used to train and test much easier than from scratch.

7 Difficulties

As the study focused on claiming that the dropout helps the neural network models from overfitting, there existed some difficulties or drawbacks regardless of the improved performance.

7.0.1 Increased training time

Dropout regularization increases model training time because the model needs to train for a long time to converge. In some cases, more iterations than other regularization methods may be required for dropout to converge. However, this drawback can be mitigated by using techniques such as batch normalization.

7.0.2 Increased test time

Dropout requires multiple passes through the network during testing to obtain predictions. This can increase model inference time. However, the computational overhead during testing can be reduced by computing a single forward pass through the network and scaling the network weights by the holding probabilities.

7.0.3 Difficulties in interpreting weights

Dropout regularization makes model weights difficult to interpret. This is because the weights have no meaningful interpretation as they are randomly zeroed during training. However, the purpose of dropout is to prevent overfitting and improve generalization. Although this was not a major concern as these drawbacks did not outweigh the performance enhancement it brought up to the models.

8 Discussion

In conclusion, this project investigated the effectiveness of dropout on various vision tasks and found that dropout can indeed help prevent overfitting in large neural networks. Our study was based on the replication of the paper by Srivastava et al., which demonstrated the benefits of dropout in achieving state-of-the-art results on benchmark datasets. Despite the limited availability of the code used in the original study, we were able to implement and tune the model hyperparameters to obtain similar results. Our experiments were conducted on popular datasets such as MNIST and SVHN, demonstrating the generalizability of dropout across different tasks. Overall, our findings suggest that dropout is a valuable tool for preventing overfitting in large neural networks and can improve performance on various tasks. Future directions could be to further improve hyperparameters. Another further step would be to see if these results could also be generalized/replicated across different types of datasets, such as text classification.

9 Statement of Contributions

Alexandra Ananthalekshmy: Data visualization, code review, Report

Eren: Model coding

References

[1] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15.1 (2014): 1929-1958.