Analysis of Regularization of CNNs with Cutout

Xinran Li[1], Navid Mahdizadeh Gharakhanlou[2] and Saad Benslimane[3]

***Abstract*— This paper analyze the work of DeVries *et al.*[1] on the regularization of different convolutional neural networks for the task of image classification. The main goals are to reproduce the subset of baseline model's performance, applying the baseline models to other datasets, and investigating the effect of different components in the baseline architecture. We also investigatively tune the hyper-parameters which attempted to improve the baseline model's performance.**

I. INTRODUCTION

The Convolutional Neural Network (CNN) is one of the most frequently used deep neural networks which has an admirable performance in machine learning issues, particularly in image classification data set (Image Net).[2] Nevertheless, because of the model capacity and full-connectivity, they are prone to overfitting data and require sufficient regularization techniques to generalize properly. In the paper work of DeVries *et al.*[1], a basic regularization technique called Cutout is used to increase the robustness and overall performance of CNNs. Cutout regularization technique randomly mask out square portions of input during training, while the cutout's principal advantages and capabilities are significant: 1) it is simple to implement, and 2) it can be combined with other types of data augmentation and regularization techniques to increase the performance of model even further.

Furthermore, as an interpretation to the Cutout method, an extension of dropout is introduced, which stochastically drops neuron activations during training and as a result discourages the co-adaptation of feature detectors. In contrast to normal drop-out process, the paper applied the dropout only at the input layer of CNN, and dropped out contiguous sections of inputs instead of individual pixels. In this fashion, it applies a spatial prior to dropout in input space with the final images don't have trace of dropped out area. It benefits the network to better utilize the full context of image, rather than replying on the presence of a small set of specific visual features.

[1]Xinran Li (260774237)
[2]Navid Mahdizadeh Gharakhanlou (261061899)
[3]Saad Benslimane (261031789)

The task completed in this paper are three-folds. First, the subset of baseline models' performances are reproduced, and applied on other data sets. Then, we investigate the effects of components of baseline's ResNet18 by ablation study, including tuning and replacing model itself to achieve more convinced conclusion. Lastly, the experiment visualize and analyze extensively-tuned hyper-parameters of baseline's ResNet18, with the goal of surpassing the original paper's performance.

II. RELATED WORK

A. Investigation on Models from Paper

To evaluate the efficiency of cutout regularization technique, DeVries *et al.*[1] uses ResNet18, WideResNet architectures on a variety of popular image data sets where Table VI has illustrated the properties of these data sets altogether. Besides, several preprocessing tasks have been done on aforementioned data before feeding into the models, which are shown in Figure 1. Furthermore, Table VII implicitly describes the hyper-parameters of different models used in paper including dropout probability and size of cut-out region. It should optimized cutout size region is achieved by splitting training dataset into two groups of training and validation with portion of 9:1.

The original paper by DeVries *et al.* compares the error rates with models applying on several data sets in existence and non-existence of cutout and/or data augmentation which its subset of result are shown in Table I. The paper concludes that CNNs with cutout and data augmentation yields less error rates on image classification tasks than without cutout or/and data augmentation. Besides, the paper claims that cutout yields these performance improvements even when applied to complex models that already utilize batch normalization, dropout, and data augmentation. To obtain this conclusion, same models are re-runed on reported datasets to confirm paper's statement. The reproduced results are displayed below in Table II, which are aligned with paper's results in Table I.

TABLE I: Test error rates (%) in paper, "+" beside the data name indicates data augmentation

Models	$C10$	$C10+$	$C100$	$C100+$
ResNet18	10.63 ± 0.26	4.72 ± 0.21	36.68 ± 0.57	22.46 ± 0.31
ResNet18 + cutout	9.31 ± 0.18	3.99 ± 0.13	34.98 ± 0.29	21.96 ± 0.24
WideResNet	6.97 ± 0.22	3.87 ± 0.08	26.06 ± 0.22	18.8 ± 0.08
WideResNet + cutout	5.54 ± 0.08	3.08 ± 0.16	23.94 ± 0.15	18.41 ± 0.27

TABLE II: Reproduced test error rates (%) in paper, "+" beside the data name indicates data augmentation

Models	$C10$	$C10+$	$C100$	$C100+$
ResNet18	11.13	4.83	36.3	21.67
ResNet18 + cutout	10.17	4	33.63	21.41
WideResNet	6.97	3.94	26	19.58
WideResNet + cutout	5.59	2.96	24.06	18.49

B. Comparison with Other Data sets

While the paper work by DeVries *et al.* has only make use of C10, C100 and SVHN, results from other data sets are examined in order to get a more convinced results about paper's conclusion on Cutout. Thus the performance of baseline model ResNet18 on Fashion-MNIST and MNIST data sets are examined for comparable result, which are displayed in Table III.

TABLE III: Test error rates (%) on MNIST and Fashion-MNIST, "+" beside the data name indicates data augmentation (mirror + crop)

Models	Fashion-MNIST	Fashion-MNIST+	MNIST	MNIST+
ResNet18	6.12	4.9	0.8	0.3
ResNet18+cutout	5.79	4.85	0.7	0.3

As seen from this table, RestNet-18 performs better on Fashion-MNIST and MNIST with less errors (6.12% for Fashion-MNIST and 0.8% for MNIST). These errors decrease when using data augmentation and cutout regularization, to reach 4.85% for Fashion-MNIST and 0.3% for MNIST.

The reason why ResNet18 performs better on these data sets is due to the less complexity of these images, which makes model easier to converge on a better solution. Moreover, it shows that the performance improvement of Cutout on these datasets stay aligned with paper's conclusion.

III. Experiment and Result

A. Ablation Study

We conduct ablation study in CIFAR-100 data set using the same settings in Table VII. We focus on the regularization components on baseline's ResNet18, explore their effects on performance. Besides, we replace the ResNet18 to obtain the effects of model itself on result.

Data Augmentation

From previous results with ResNet18 in Table II, there's an significant improvement of error rate from 36.3% to 21.67% (improved 14.63%) by using of data augmentation without cutout. While with existence of cutout, the data augmentation improved error rate from 33.63% to 21.42% (improved 12.22%), which are slightly less than the improvement without cutout.

It's also inspected that data augmentation even improves performance on complex models as WideResNet, but since the model itself performs higher accuracy, the improved performance is not as remarkable as in simple model like ResNet18.

Cutout

As the performance on ResNet18 in Table II shown, there's an improvement of error rate from 36.3% to 33.63% (improved 2.67%) using Cutout as an extension of dropout in input space, without data augmentation. While with data augmentation, the error rate improved from 21.67% to 21.41% (improved 0.26%). It's observed that after use of data augmentation, the Cutout method less likely improved the as significant as before.

Compared to improvement by using of data augmentation from previous, the improvement of cutout seems even less notable. However, the result still draws attention to the fact that Cutout regularization improves model performance of even complex models like WideResNet in a similar scale. The original paper spells out a more detailed observation that shallow layers of network experience a general increase in activation strength, while in deeper layers, we see more activations in the tail end of distribution.

Cutout Size vs. Number of Class

We evaluate Cutout size with {8, 16} on C10 and C100 by using ResNet18 model with cutout and data augmentation, results are shown in Fig 9. It seems as number of classes increase, the smaller cutout region performs better. The paper concludes this observation that "*this makes sense since more fine-grained detection is required then the context of the images will be less useful for identifying the category.*"[1] Instead, smaller and more nuanced details are more important and too large cutout region would have under-fitting the model.

Model Selection

An comparable result between using ResNet18 and WideResNet is displayed in Table II, where by using more complex models will generally increase the performance. In order to achieve more convinced

result, the performance on ResNet50 with the same hyper-parameters as ResNet18 are examined on those data sets mentioned in DeVries's paper, along with MNIST and Fashion-MNIST. The result is shown in Table IV.

TABLE IV: Test error rates (%) with ResNet50, "+" beside the data name indicates data augmentation (mirror + crop)

Models	C10	C10+	C100	C100+	Fashion MNIST	Fashion MNIST+	MNIST	MNIST+
ResNet50	10.2	4.8	34.4	21.6	6.23	5.06	0.3	0.7
ResNet50 + cutout	8.6	4.2	31.3	20.41	4.94	4.75	0.3	0.6

By comparing Table IV with Table II and Table III, we conclude that by training ResNet model with more layers generally improves the accuracy by 0.3-2% depending on the dataset and regularization technique used. For example compared to both ResNet18 and ResNet50 with data augmentation and cutout regularization, ResNet50 lessens error rate from 21,41 to 20,41% for C100, from 4,85% to 4,75% for Fashion-MNIST and from 0,7% to 0,6% for MNIST. Moreover, the result shows that Cutout can work in conjunction with complex model which improves the accuracy even already utilize with data augmentation.

B. Improve Baseline Approach

We conduct experiment that objective is to surpass baseline performance OF ResNet-18 on CIFAR-10, we extensively tune the model for various choices of hyper-parameters and determine the optimized value for them. Due to a limited time and GPU, we only make experiment on ResNet18 for CIFAR-10 dataset.

We focus on the hyper-parameters of learning rate, batch size, and cutout size and as changing factors during the training, in order to reach a more stable increasing trend of accuracy. As shown in Figure 2, Figure 3, and Figure 4, where the accuracy trend are plotted as graphs with three changing factors. We observed the optimized learning rate is 0.025 (without weight decay), optimized cutout size is 10, and optimized batch size is 50 for running 200 epoch, while other hyper-parameters stay the same.

We also compare training and testing accuracy of the optimized ResNet18 we have determined with the baseline mentioned in the paper in four diverse cases (Figure 5, 6, 7, 8). We observe from these figures that throughout training on optimized ResNet18 with optional Cutout and/or data augmentation, the accuracy is more stable over time (less variations than before) which helps the model to converge to a solution faster.

The table V compares the errors obtained from four diverse cases by result of reproduced baseline and result of optimized model. As seen the accuracy has generally improved by 0.3 - 0.5% in three cases using the optimal values. For ResNet18 with Cutout and data augmentation, the accuracy is already optimized thus no improvement as an exception.

TABLE V: Comparable result of Reproduced Error Rates (%) and Optimized error rates (%) on CIFAR-10

	ResNet18		**ResNet18+Cutout**	
	C10	C10+	C10	C10+
Reproduced	11.13	4.83	10.17	4
Optimized	10.61	4.74	9.87	4

IV. Discussion

Through this project, we managed to build a deeper understanding on the cutout regularization on CNNs. There are several key takeaway as we pursued the reproduction and ablation study:

- Cutout works complementary to the existing forms of dropout and data augmentation on different models, although improvements are not as impressive as data augmentation.
- Cutout achieves state-of-art performance both on simple dataset as MNIST and complex dataset as CIFAR-100; it performs more significant on simple dataset since more fine-grain detection needed for more classes dataset.
- Cutout works effectively regardless of simple or complex modern architecture such as ResNet18 and ResNet50. However, CNNs with more layers generally reach higher accuracy on large dataset.
- There is an notable exception that for training complex CNNs on simple image dataset like MNIST, where cutout provides no improvement on the performance.

V. Conclusion

In this work, we explore the robustness of cutout regularization, reproduce and analyze the paper's result, and contribute an optimized ResNet18 performs $\tilde{0}$.4% higher than baseline. The main challenge is to improve the model performance, since the cutout performs less effective on more number of classes, plus It's general but non-efficient approach by building more layers of CNNs to achieve better result. We only improve the ResNet-18 on CIFAR-10, and this challenge stated are not investigated. For the future approach, we are curious on other MLP approach like vision transformer and how Cutout could alternatively apply on it.

REFERENCES

[1] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[2] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, Ieee, 2017.

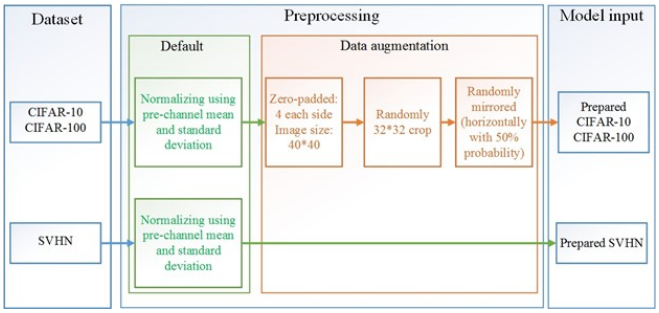

Fig. 1: Preprocessing tasks on baseline models.

Fig. 2: Accuracy of ResNet18 on CIFAR-10 vs. Learning rate

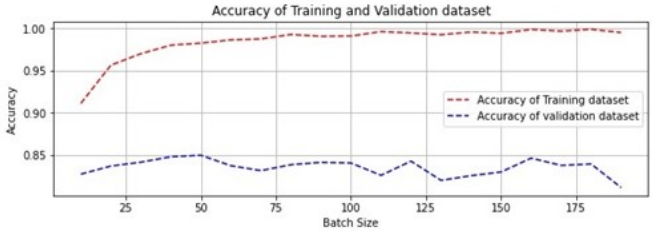

Fig. 3: Accuracy of ResNet18 on CIFAR-10 vs. Batch size

Fig. 4: Accuracy of ResNet18 on CIFAR-10 vs. Cutout size

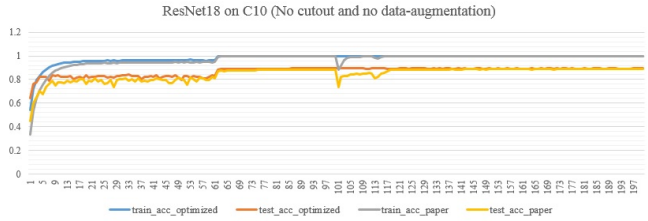

Fig. 5: Epochs vs. Accuracy (ResNet18 on C10 without cutout and without data-augmentation)

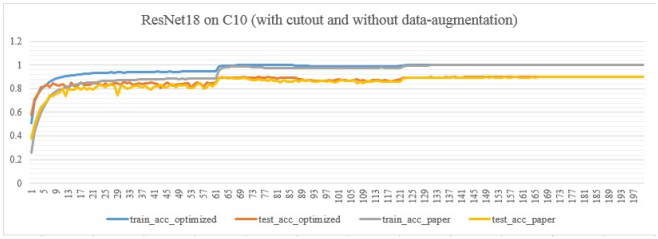

Fig. 6: Epochs vs. Accuracy (ResNet18 on C10 with cutout and without data-augmentation)

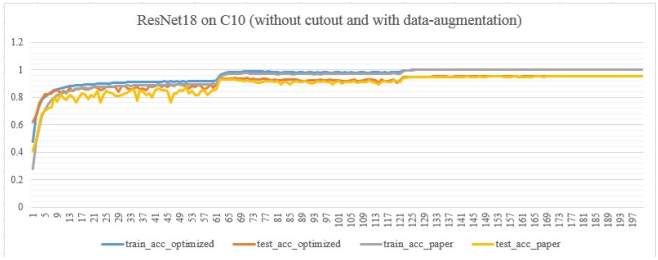

Fig. 7: Epochs vs. Accuracy (ResNet18 on C10 without cutout and with data-augmentation)

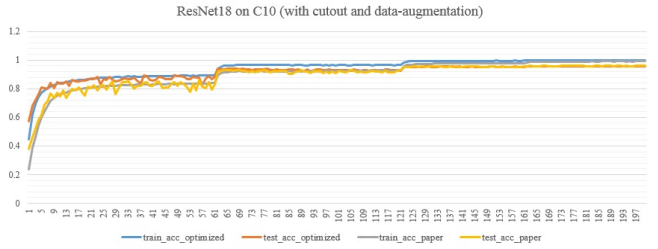

Fig. 8: Epochs vs. Accuracy (ResNet18 on C10 with cutout and data-augmentation)

TABLE VI: The properties of data in DeVries *et al.*[1]'s paper

Dataset	Batch Size	Size	Classes	Training Size	Testing Size
CIFAR-10	60,000	32*32	10	50,000	10,000
CIFAR-100	60,000	32*32	100	50,000	10,000
SVHN	630,420	32*32	10	604,388	26,032

TABLE VII: The hyper-params of models in DeVries *et al.*[1]'s paper

Dataset	Models	Epochs	Batch Size	Optimizer	Dropout Probability	Optimized Cutout Size
CIFAR-10 **CIFAR-100**	ResNet18	200	128	SGD momentum: **0.9** weight decay: **5e-4** learning rate: **0.1** decreases by a factor of **5x** at **60th, 120th, 160th** epochs	0.3	16*16 (C10) 8*8 (C100)
CIFAR-10 **CIFAR-100**	WRN-28-10 (depth: 28 widening factor 10)	200	128	SGD momentum: **0.9** weight decay: **5e-4** learning rate: **0.1** decreases by a factor of **5x** at **60th, 120th, 160th** epochs	0.3	16*16 (C10) 8*8 (C100)
SVHN	WRN-16-8 (depth: 16 widening factor 8)	160	128	SGD momentum: **0.9** weight decay: **5e-4** learning rate: **0.01** decreases by a factor of **10x** at **80th, 120th** epochs	0.3	20*20

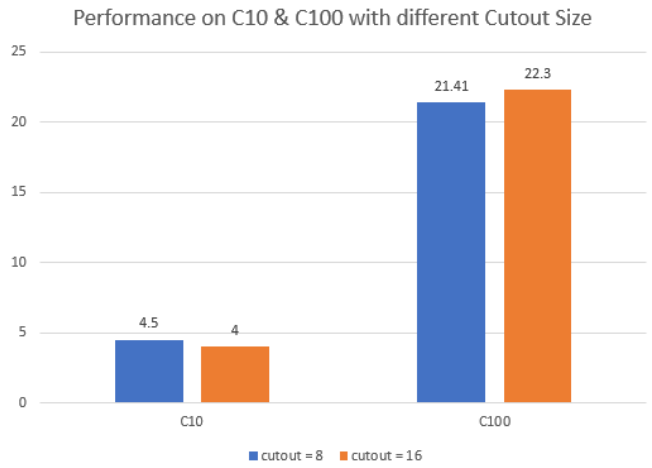

Fig. 9: Performance on C10 & C100 with different Cutout Size