ECE 590-03 Final Project Report
Project 1: Improved Regularization of Convolutional Neural Networks

Xian Sun
Electrical and Computer Engineering Department
Duke University
xs75@duke.edu

Abstract

Overfitting is a common problem for Convolutional neural networks (CNN). This project is mainly exploring the combination of two regularization techniques: mixup and cutout in order to improve the generalization of CNN models. We explore performance of them on CIFAR10-Resnet with different depth and try to find the best combination of them on CIFAR10-ResNet18. And then we transfer this combination on other datasets like SVHN and Fashion-MNIS. In general, the experiments show that both of mixup and cutout works well on addressing overfitting, but it is harder for mixup to handle overfitting problems if the model goes deeper. And the best combination of them also works well for these two other datasets.

1 Introduction

Recently deep learning has been a powerful tool in computer vision and it is applied widely in our daily life. It does a good job on human face recognition, object detection, etc. But sometimes overfitting is still a big problem for deep learning techniques, especially for some critical decisions. For instance, if there is an error in human face recognition at police department, which may lead policeman to mistreat some innocent people as criminal. Thus, it is very important for us to explore a powerful regularization techniques to solve overfitting problem.

In this project, we are trying to explore mainly the combination of two regularization techniques: cutout and mixup to address overfitting problems on Resnet. We found out that these two methods have very little computational cost and can solve overfitting effectively for Resnet on CIFAR10 , SVHN and Fashion MNIST. At first I Implemented cutout and mixup to address overfitting problems for ResNet18,34 and 50. Then I did grid search for a good combination of cutout and mixup for ResNet18-CIFAR10 and transfer the best combination of cutout and mixup to SVHN and Fashion MNIST, trained on ResNet18

2 Background

Nowadays, convolutional neural networks are doing well on object detection, image classification, etc. But they are easily exposed to a common problem: overfitting, especially for the model with large depth. Thus, it is important to find some powerful regularization techniques to improve the generalization of CNN models so that our CNN models can do a better job on unseen dataset. A. Krizhevsky et al. proposed data augmentation approach on ImageNet in 2012 [1]. In 2017, Terrance DeVries et al. found an improved regularization method named cutout, which forces models to focus

ECE 590-03 (2020 Fall) final report.

on the full image instead of some tiny details [2] . In 2018, Moustapha Cisse et al. designed a method named mixup, which encourages model to behave linearly in-between training examples [3].

Both of these methods can work well on overfitting problems individually. But in this project, we are interested in how these two methods work together to have a better performance.In order to improve the performance of CNN model, we also want to try to find a good combination of them on CIFAR10 and transfer it into datasets like SVHN and Fashion-MNIST.

3 Method

3.1 Cutout

The main idea of cutout is that we mask some fixed-size area of the input images. To implement this, we can simply apply a fixed-square mask on the random location of every image before training. We also call the area being masked patch. The simple intuition of cutout is that it encourages models to focus on the full image and pay less attention on some details in the image. There are two parameters in this method: one is the number of patches and the other is the size of one patch . The main steps are below.

· **step1** Initialize the number of patches = N and length = L

· **step2** Randomly select N patches with fixed-size L*L in the image.

· **step3** Assign 0 into all selected patches.

3.2 mixup

The main idea of mixup is to create a new image by mixing two images drawn from the training data, according to some proportion drawn from beta distribution. To implement Mixup, we simply do data transformation during each batch of training based on equation1 and equation2.

$$image_{new} = \lambda * image_i + (1 - \lambda) * image_j \tag{1}$$

$$label_{new} = \lambda * label_i + (1 - \lambda) * label_j \tag{2}$$

where λ is the proportion drawn from Beta distribution with parameter (α, α). You can also use $numpy.random.beta$ to generate data drawn from beta distribution in python. And $(image_i, image_j)$, $(label_i, label_j)$ are pairs of input image and label, drawn from training dataset randomly, and $image_{new}$, $label_{new}$ are new images and labels that will be put into model for training.

3.3 Convolutional neural networks

In the first experiment, I explored the relationship between the performance of these two techniques and the depth of Resnet model. In the second experiment, Grid search for a good combination of cutout and mixup for Resnet18-CIFAR10. In the third experiment, Transfer the best combination of cutout and mixup to SVHN and Fashion MNIS. All code for those three experiments are referred to cutout paper[2] and mixup paper[3]. All experiments are following these hyper parameter settings.

· **Models:** ResNet18, ResNet34, ResNet50

· **Training epochs:** 100

· **Batch size:** 128(train), 100(test)

· **Learning rate:** start at 0.1, decay by 20% at 30, 60, 80 epochs

· **Optimizer:** SGD with momentum 0.9

· **L2 regularization:** weight decay 5×10^{-4}

· **Data augmentation:** Randomcrop(32, padding=4), RandomHorizontalFlip(), normalization

4 Experiment results

4.1 Experiments on CIFAR10-Resnet18,34,50

With fixed $\alpha = 0.1$ for mixup and holes $= 1$ and length $= 16$ for cutout, I tested Resnet18,34,50 respectively. From the figure 1, the baseline is the traning process without mixup and cutout where there is a big gap between traning and testing curves. It means that the model is overfitting in baseline. But for the training process with cutout, we can find the gap between training and testing accuracy curves become smaller which means the overfitting is reduced. For the training process with mixup, we can see two curves are overlapping together which means that overfitting is solved completely with this method. All results show that mixup and cutout perform well on overfitting individually.

From the figure 1, for mixup, in the three figures on the third column, we can find the gap between training and testing is becoming slightly larger as we improve the depth of Resnet, which means it is harder for mixup to solve overfitting problems as the Resnet become deeper. I think it makes sense because as the depth of model is going large, it will reduce the influence of mixup as mixup do processing on data before the whole training process.

From the figure 1, for cutout there is not big difference between ResNets with different depths. I think it might be because of the length and the number of holes I set. In this case, the performance of cutout does not change with the depth of ResNets.

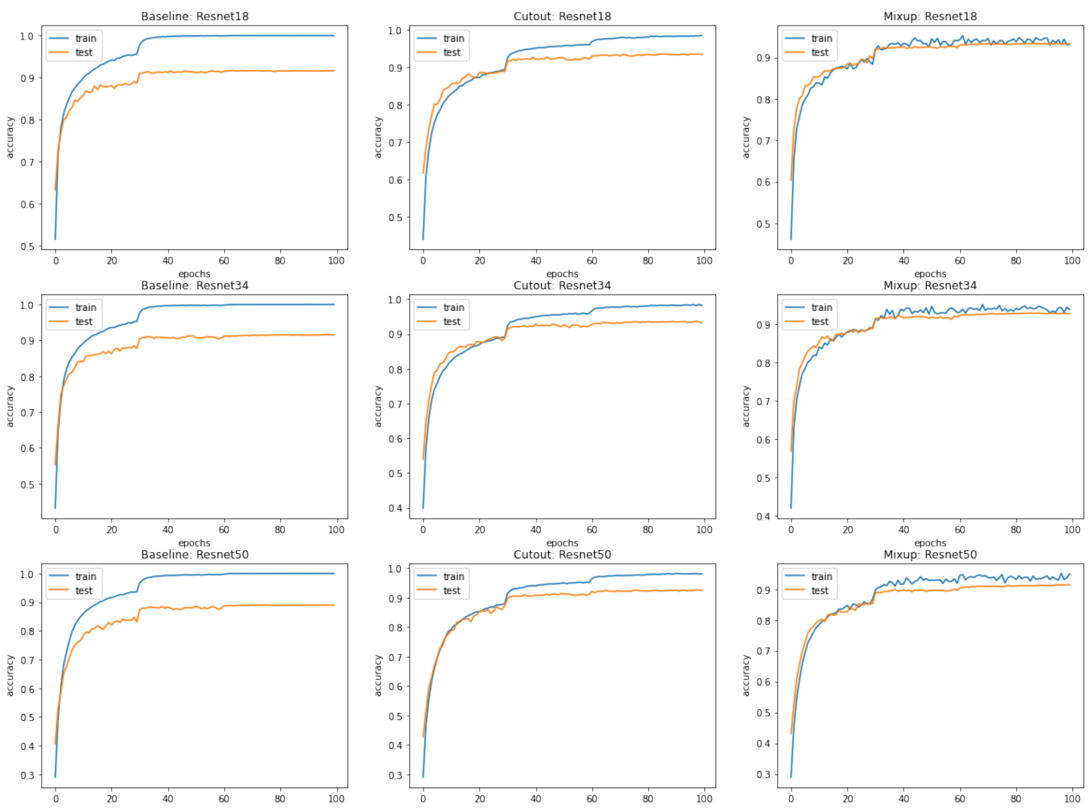

Figure 1: Training process for Resnet18,34,50 with baseline, cutout and mixup respectively

4.2 Grid search for a good combination of cutout and mixup for Resnet18-CIFAR10

Based on CIFAR10, I did grid search on ResNet18 with α from $[0.1, 1, 2]$ for mixup and length from $[8, 10, 12, 16, 18, 20]$and tried to find the best combination where we should have the smallest testing error. From the figure 2, , we can notice that ResNet18 has the smallest test error 5.48% when I set the combination where patch length = 10 and $\alpha = 1$.

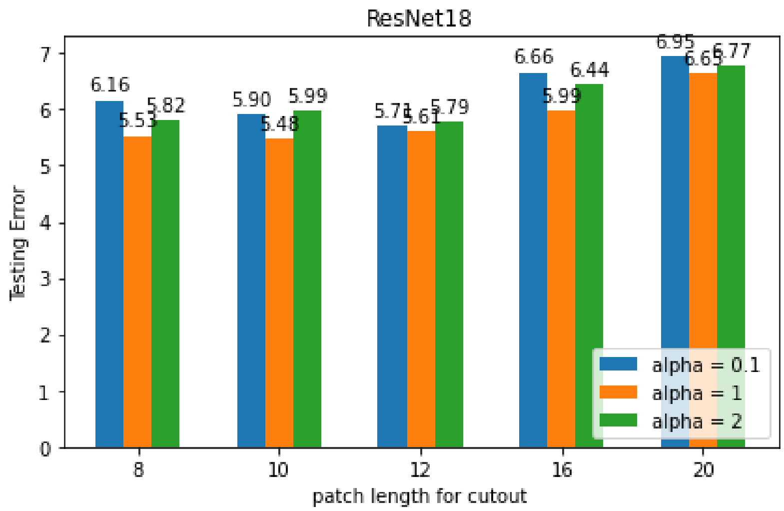

Figure 2: Grid search for a best combination of length for cutout and alpha for mixup.

4.3 Transfer the best combination of cutout and mixup to SVHN and Fashion MNIST

I trained ResNet18 on Fashion-MNIST and SVHN with the best combination of mixup and cutout from the previous experiment. From the figure 3 , we can see after I transferred the best combination of mixup and cutout, the test accuracy is improved by 1% for Fashion-MNIST and 3% for SVHN. Besides, we can see that this mixture of two techniques also make the objective curve more stable. In general, I think this combination of mixup and cutout perform well on fashion-MNIST and SVHN datasets.

It makes sense that cutout and mixup still works well for those two datasets but we cannot make sure this combination is still best one for those two datasets. Thus, if we want to have the highest test accuracy for those datasets, it is better for us to do grid search based on different dataset to make sure the combination I choose can achieve the highest test accuracy.

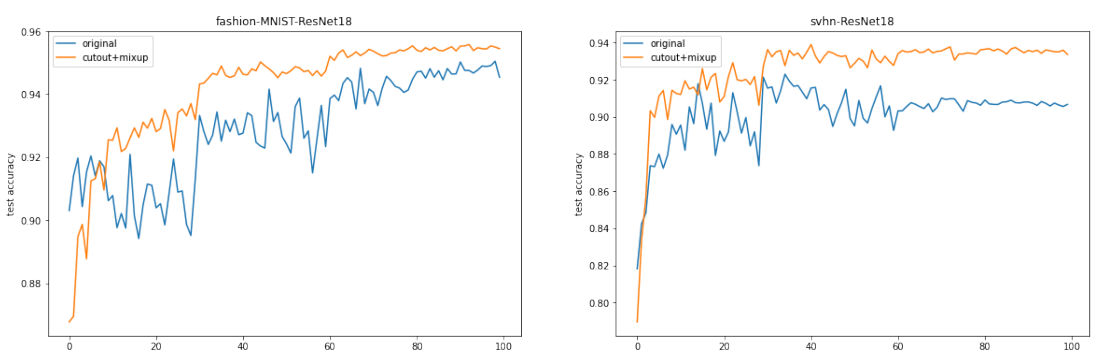

Figure 3: Testing accuracy curves during training process for original and with the combination of mixup and cutout for Fashion-MNIST and SVHN datasets

5 Conclusions

Cutout and mixup are two regularization techniques that can work well together to solve overfitting problems for ResnNet effectively on datasets like CIFAR10, SVHN and Fashion MNIST.

References

[1] Alex Krizhevsky and Ilya Sutskever and Geoffrey E. Hinton. *ImageNet Classification with Deep Convolutional Neural Networks*. 2012.

[2] Terrance DeVries and Graham W. Taylor. *Improved Regularization of Convolutional Neural Networks with Cutout*. 2017.

[3] Terrance Hongyi Zhang and Moustapha Cisse and Yann N. Dauphin and David Lopez-Paz. *mixup: Beyond Empirical Risk Minimization*. 2018.

A Timeline and task allocation

· **First week:** Read papers and implemented mixup and cutout.

· **Second week:** Trained Resnet18,34,50 with mixup or cutout respectively.

· **Third week:** Did grid search for a good combination of cutout and mixup for Resnet18-CIFAR10.

· **Fourth week:** Transfer the best combination of cutout and mixup to SVHN and Fashion MNIST.

B (Optional) Implementation detail

You may put implementation details like hyperparameter choices here if there's no space in the main article. Be sure to mention this in the mean article if you do so.

C (Optional) Additional experiment results

You may put additional figures or tables here to cover more detailed results than what you have in the main article. Make sure to clearly refer to the results in appendix in the main article. Note that reviewers are not required to read the appendix.