# Mini-Project 1: Residual Network Design

**Congkai Geng, Zhemian Li, Lanxin Hu**
cg3946@nyu.edu, zl3985@nyu.edu, lh3222@nyu.edu
ECE, NYU Tandon

## Abstract

In this project, a ResNet architecture is designed to maximize the accuracy on the CIFAR-10 dataset under a constraint of less than 5M trainable parameters. ResNet hyper-parameters, N, $B_i, C_1, F_i, K_i$ and P are tuned. In model training, some strategies are used, such as data augmentation, optimizer, regularization scheme. Our code and models are available at: https://github.com/CCGV2/DLProject1

## 1 Introduction

Residual networks were widely used for image classification in modern computer vision applications. In this project, we designed and trained our own ResNet model for CIFAR-10 image classification. Our goal is to maximize accuracy on the CIFAR-10 benchmark while keeping the size of the ResNet model under budget. Model size, typically measured as the number or trainable parameters, is important when models need to be stored on devices with limited storage capacity, mobile devices for example.

## 2 Methodology

### 2.1 ResNet hyper-parameters selection

To meet the constraints that the maximum number of parameters should be 5M, we decreased the in_planes from 64 to 42.

### 2.2 Batch size

In the training process, we used batch size of 64 and 128. To improve the converging speed, we first increased the batch size from 64 to 128. The experimental result (Figure 1) shows that increasing the batch size does not have strong effect to the convergence speed. As the accuracy of the model with batch size of 128 is better in the early stage, we decided to use set the batch size to 128.

### 2.3 Learning Rate

Obviously, as shown in class, the magnitude of learning rate is important, it will decide whether the model can get the decision point. Large learning rate may cause diverge and small learning rate may cause sub-optimal solution. However, the learning rate should reduce as well, which is also shown in class, otherwise, the model will likely bounce near the convergent point. So, we tried a learning rate scheduler.[1]

**Cosine Annealing**

Within the i-th run, the learning rate is defined as follows:

$$\eta_t = \eta^u_{min} + \frac{1}{2}(\eta^i_{max} - \eta^i_{min})(1 + cos(\frac{T_{cur}}{T_i}\pi)), \tag{1}$$

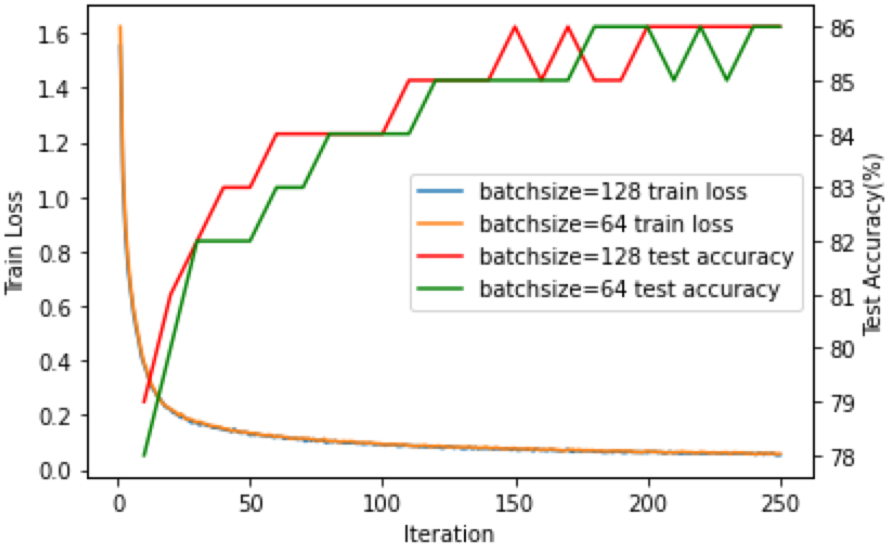

Figure 1: Batch size: 64 vs. 128

where $\eta^{i}_{min}$ and $\eta^{i}_{min}$ are ranges for the learning rate, and $T_cur$ accounts for how many epochs have been performed.[5]

This make the learning rate become a decreasing value and will become very small when the training is almost end.

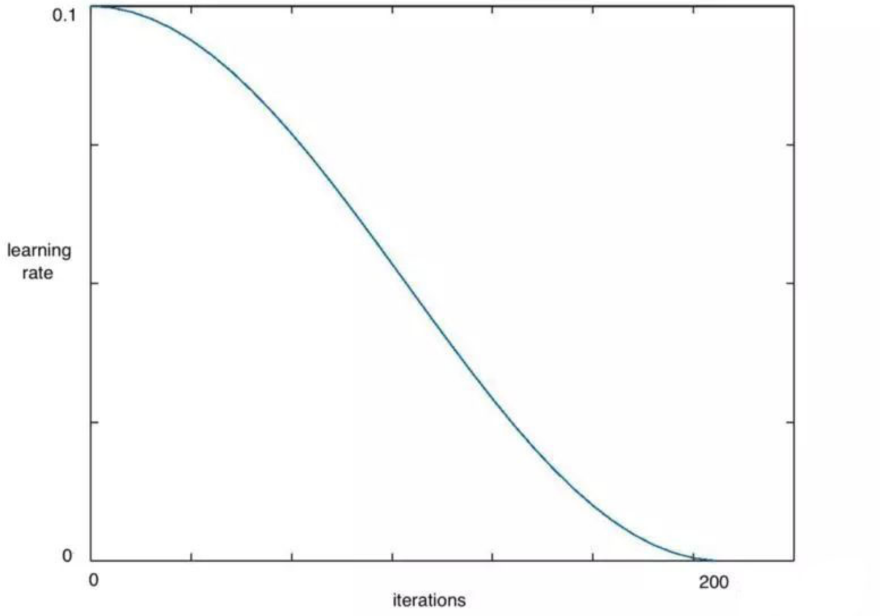

Figure 2: learning rate

### 2.4 Data augmentation

Data augmentation is an effective data preprocessing technique for improving the accuracy of modern image classifiers. ResNets heavily relies on big data to avoid overfittiing, data augmentation can be used to enlarge the dataset and improve the diversity of the dataset. In this project, by experimenting with different data augmentation schemes, we decided to use AugMix for our final model, since it has the best performance. The table 1 below indicates the experimental results of different data augmentation schemes.

Table 1: The test set accuracy on CIFAR-10 with different approaches for data augmentation.

| | **randomflip** | **AutoAugment** | **AugMix** |
|---|---|---|---|
| Accuracy | 87.08% | 89.34% | 95.56% |

#### randomflip

We added a preprocessing layer which randomly flips images during training. We experienced both horizontally and vertically flipping the images. Horizontally flipping images improves the accuracy, while vertically flipping images reduces the accuracy. Using horizontal random flip with the probability of 0.5, our model1 achieved the highest accuracy of 87% in 250 epochs.

#### AutoAugment

While fixed data augmentation schemes are not flexible for large dataset, in our implementation, we used AutoAugment to automatically search for improved data augmentation policies [2]. This approach consists of two components: a search algorithm and a search space. At a high level, the search algorithm samples a data augmentation policy S, which contains information about what image processing operation to use and the magnitude of the operation. After training the ResNet with the selected operation, the validation accuracy S will be sent back to update the search algorithm. In this project, we used AutoAugment approach implemented by Philip Popien and Ali Hassani[6] With the learning rate of 0.000075, the model reached the highest accuracy of 89% in 250 epochs.

#### AugMix

Augmix is a method that mix several different augmentation up with a parameter. It's similar to Mixup. First the algorithm pick 1 to 3 random augmentation and combine them with the algorithm by weight. This mixed up image requires a new loss function called Jensen-Shannon divergence. The Jensen-Shannon divergence can be understood to measure the average information that the sample revels about the identity of the distribution from which it was sampled.[3]

### 2.5 Optimizer

Optimizers are widely used in the training process to wisely adjust the weights and learning rate of neural network to minimize the loss function. In this project, we applied different optimizers to the training process, based on the accuracy, we decided to use Rectified Adam for the final model.

Table 2: The test set accuracy on CIFAR-10 with different optimizers.

| | **SGD** | **SGDR** | **Adam** | **RAdam** |
|---|---|---|---|---|
| Accuracy | 89% | 95.6% | 89% | 91% |

#### Stochastic Gradient Descent

Stochastic Gradient Descent update the parameters of the model one by one. Advantages of SGD are that it frequently update the parameters and it is useful in large dataset as it update the parameters one example at a time. The disadvantages are that it is computationally expensive and may result in noisy gradients.

### Adam

To decrease the computationally cost and memory requirements, we also used Adam for the experiment. Adam is an algorithm for first-order gradient optimization of stochastic objective functions, based on adaptive learning rates for each parameter.

### Rectified Adam

Adam uses the exponential moving average to calculate the adaptive learning rate, the variance of the learning rate is not constant during the training [4]. Therefore, we decided to use Rectified Adam, which keeps the variance of the learning rate constant.

### Stochastic Gradient Descent with Warm Restarts

However, SGD with cosine annealing breaks the status. It's a state-of-art mechanic in 2018 and can improve learning process which is discussed in Learning Rate section.

## 3 Results

Here is the performance of our final model. After training for 1200 epochs, the highest accuracy reaches 95.6%.

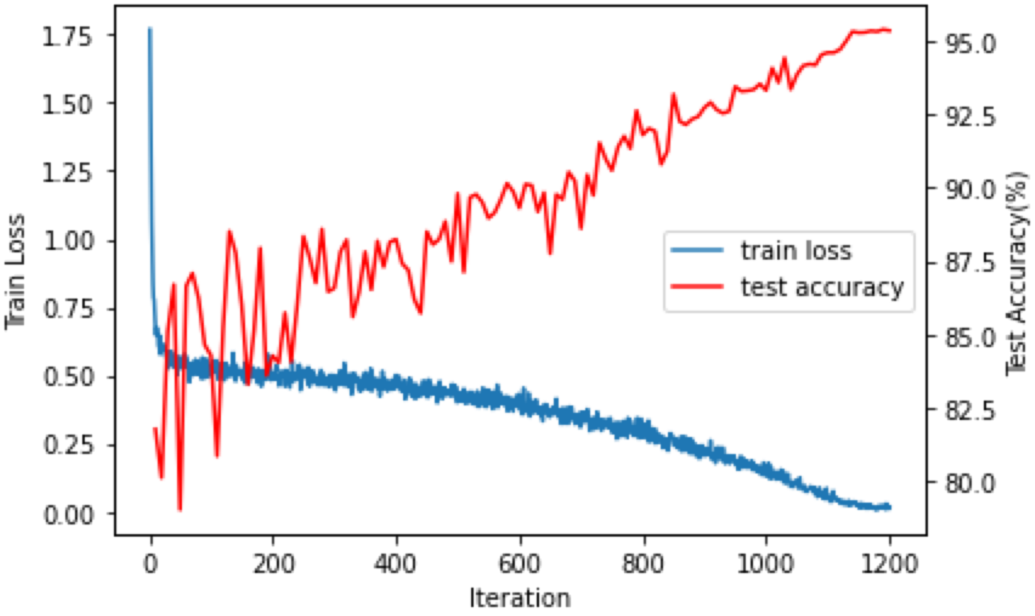

Figure 3: Loss and accuracy

## 4 Conclusion

In this project, we redesigned the ResNet-18 architecture and implemented an efficient training process. We first reduced the in_plane parameter to reduce the total number of parameters to 4813298. For the data processing technique, we used AugMix for data augmentation, which improves robustness and uncertainty measures on image classification tasks. We also used Stochastic Gradient

Descent with Warm Restart to minimize the loss. We adopted a learning rate scheduler to dynamically decrease the learning rate during the training.

## References

[1] Optimization algorithms: Schedulers.

[2] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019.

[3] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

[4] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

[5] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[6] Philip Popien. AutoAugment - learning augmentation policies from data, 2018.