Final report

Training program for Studienarbeit/Projektarbeit

Sourav Poudyal

University of Siegen

Faculty IV: School of Science and Technology

Department of Computer Science

Visual Computing Group & Computer Vision Group

May 10, 2023

Confirmation

I hereby confirm that this report is entirely my own work and that I have not used any additional assistance or resources other than indicated. All quotations, paraphrases, information and ideas that have been taken from other sources (including the Internet as well as other electronic sources) and other persons' work have been cited appropriately and provided with the corresponding bibliographical references. The same is true of all drawings, sketches, pictures and other illustrations that appear in the text. I am aware that the neglect to indicate the used sources is considered as fraud and plagiarism in which case sanctions are imposed that can lead to the suspension or permanent expulsion of students in serious cases.

Siegen 10.05.2023	Sourav Poudyal
Place, Date	Signature

Contents

1 Introduction

Current Deep Learning models gives good predictions when the test dataset has similar distribution as the training and validation data set, however if there is slight shift in the distribution than the original, the model prediction accuracy degrades, for example slight introduction of corruption or perturbation to the image in the dataset is misclassified by the model. This means the model is not robust against the changes in the dataset distribution and the prediction is uncertain to be deployed for practical applications. In this project the data Augmentation technique according to AugMix [1] is used in the CIFAR-10[2] dataset to train ResNet18 consisting of 18 layers, including 16 convolutional layers and 2 fully connected layers [3] and ConvNext Tiny consisting of 6 convolutional layers with small filter sizes (3x3 or 1x1), followed by 2 fully connected layers [4], and test is carried out in CIFAR-10C[2] and CIFAR-10P[2] dataset. In this Project Evaluation cretrion like Test Error on CIFAR 10, Mean Corruption Error(mCE) for CIFAR10C dataset and Mean Flip Probability for CIFAR-10P as desctibed in [2] and [1], is analysed to determine networks performance.

Objective of the Work

The objective of the work includes:

1. To Train the ResNet18 and ConvNext Tiny model on CIFAR-10 and evaluate on CIFAR-10, CIFAR-10C and CIFAR-10P

2. Deploying both network with:

    a. AdamW optimizer with CosineAnnealingLR learning rate scheduler.

b. and SGD optimizer with LambdaLR learning rate scheduler.

3. Adding Tensorboard implementation to log Train and Test losses and deriving conclusion form the results.

4. Hyper parameter Tuning of the model to improve its performance.

2 Experiments

2.1 Experiment 1

In this Experiment CIFAR-10 dataset that is augmented with the algorithm described in [5] is used to train both pretrained and non pretrained ResNet18 and ConveNext Tiny model with initial hyper parameter of learning rate 0.1 and weight decay 0.0005 and SGD optimizer with lambda learning rate scheduler as described in the table 1 and evaluation is done on CIFAR-10 , CIFAR 10C and CIFAR-10P model such that the Test Error is the evaluation error on CIFAR10, mCE(Mean Corruption error) is the mean evaluation error on all the corruption classes on CIFAR-10C of and mFP(mean Flip Probability) is the mean Flipping Probability on all the Perturbation classes on CIFAR-10P.

Similarly in second part of the first experiment with same learning rate and weight decay AdamW optimizer and CosineAnnealingLR was used as seen in table 2

Table 1: SGD optimizer and LambdaLR scheduler with Learning rate 0.1 and weight decay 0.0005

Plot	Model	Pretrained	Epoch	Test Error (%)	mCE (%)	mFP
a	convnext tiny	Yes	100	9.93	16.132	0.02087
b	convnext tiny	No	100	90.00	90.00	0.00000
c	resnet18	Yes	100	11.83	17.492	0.02335
d	resnet18	No	100	11.01	16.755	0.02225

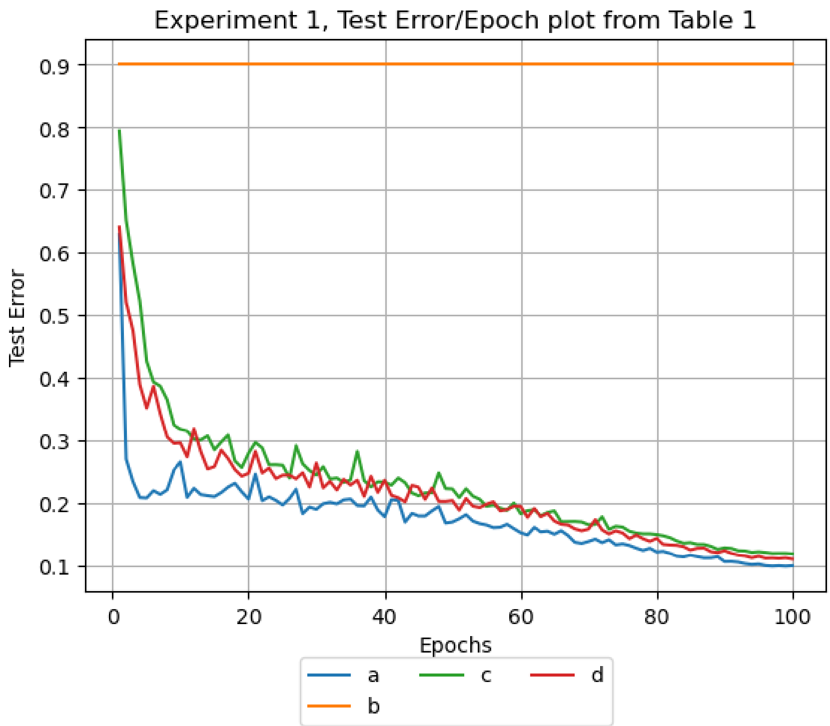

Figure 1: Test Error at each Epoch plot from table 1

Table 2: AdamW optimizer and CosineAnnealingLR with Learning rate 0.1 and weight decay 0.0005

Model	Pretrained	Epoch	Test Error (%)	mCE (%)	mFP
convnext tiny	Yes	100	90.00	90.00	0.00000
convnext tiny	No	100	90.00	90.00	0.00000
resnet18	Yes	100	14.51	20.333	0.02845
resnet18	No	100	32.10	37.408	0.04189

2.2 Experiment 2

Following the first experiment, second experiment was conducted by decreasing the initial learning rate to 0.001 and weight decay to 0.0002 with SGD and LambdaLR scheduler only on pretrained and non pretrained resnet18, however the model seem to degrade in its performance in case of resnet18 as seen in tabel 3.

Table 3: SGD optimizer and LambdaLR scheduler with Learning rate 0.001 and weight decay 0.0002

Model	Pretrained	Epoch	Learning Rate	Wight decay	Test Error (%)	mCE (%)	mFP
resnet18	Yes	100	0.001	0.0002	14.03	19.899	0.02867
resnet18	No	100	0.001	0.0002	22.49	28.238	0.03657

In second part of Experiment 2, the convnext tiny model was trained with AdamW optimizer and CosineAnnealingLR by decreasing learning rate from 0.1 to 0.01 and the 0.001.It was observed that even for learning rate of 0.01 the pretrained model produced invalid results this could be because of higher learning rate and initialization of weights the gradients seems to suffer exploding gradient problem[6], however for afterwards for non pretrained model at 0.01 and other lower learning rate the network seems to train and give good performance than in Experiment 1, furthermore in then weight decay is also decreased from 0.0005 to 0.0002 and so on until the network gives better result as seen in table 4.

Table 4: Experimnet 2 Table 2

Model	Pretrained	Epoch	Learning Rate	Wight decay	Test Error (%)	mCE (%)	mFP
convnext tiny	Yes	100	0.01	0.0005	90.00	90.00	0.0000
convnext tiny	No	100	0.01	0.0005	14.35	19.529	0.02543
convnext tiny	Yes	100	0.001	0.0005	9.38	15.680	0.01957
convnext tiny	No	100	0.001	0.0005	21.09	26.106	0.03085
convnext tiny	Yes	100	0.001	0.0002	8.51	14.852	0.01962
convnext tiny	No	100	0.001	0.0002	21.14	16.291	0.03129
convnext tiny	No	100	0.0001	0.00001	26.05	31.402	0.03263
convnext tiny	Yes	100	0.0001	0.00001	7.65	13.071	0.01995
convnext tiny	Yes	100	$1e^{-}4$	$1e-8$	8.72	14.945	0.02085

2.3 Results

2.3.1 Results of experiment 1

From table 1 it can be observed that at the hyperparameter of Experiment 1, the Pretrained convnext tiny and resnet18 model outperforms not pretrained model while convnext tiny pretrained model gives the best result on all dataset, furthermore non pretrained convnext tiny model seems to not train on the dataset at this setting which could be because of higher learning rate, exploding gradient problem[6]. Furthermore, from table 2 it can be observed that with AdamW optimizer and CosineAnnealingLR with same learning rate and weight decay performance of resnet18 seems to degrade than with previous optimiser and scheduler.

2.3.2 Results of experiment 2

As could be seen from table 5, the pretrained convnext tiny model gives best result when initially the learning rate and weight decay is lower and ratio between them is around 10 % with AdamW optimizer and CosineAnnealingLR scheduler. Although the Test error and Mean corruption erro(nCE) decreased the Mean Flip Probability has slightly decreased. Also from Figure 2 and 1 it is clear that at around 96 epoch the model achieves the desired accuracy so training beyond this epoch might lead to over-fitting or unnecessary computation cost.

Table 5: Convnext-Tiny model with AdamW optimizer and CosineAnnealingLR scheduler

Plot	Pretrained	Epoch	Learning Rate	Wight decay	Test Error (%)	mCE (%)	mFP
e	Yes	100	0.001	0.0005	9.38	15.680	0.01957
f	Yes	100	0.001	0.0002	8.51	14.852	0.01962
g	Yes	100	0.0001	0.00001	7.65	13.071	0.01995
h	Yes	100	$1e^{-}4$	$1e-8$	8.72	14.945	0.02085

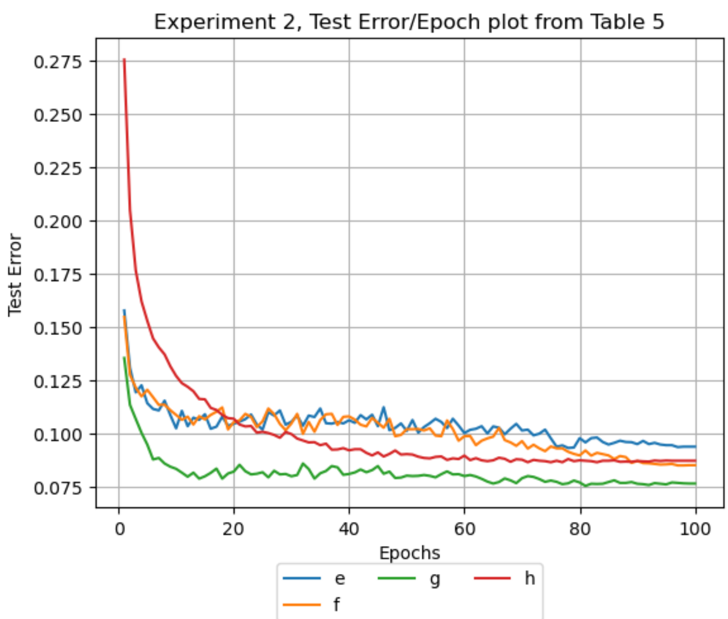

Figure 2: Test Error at each Epoch plot from table 5

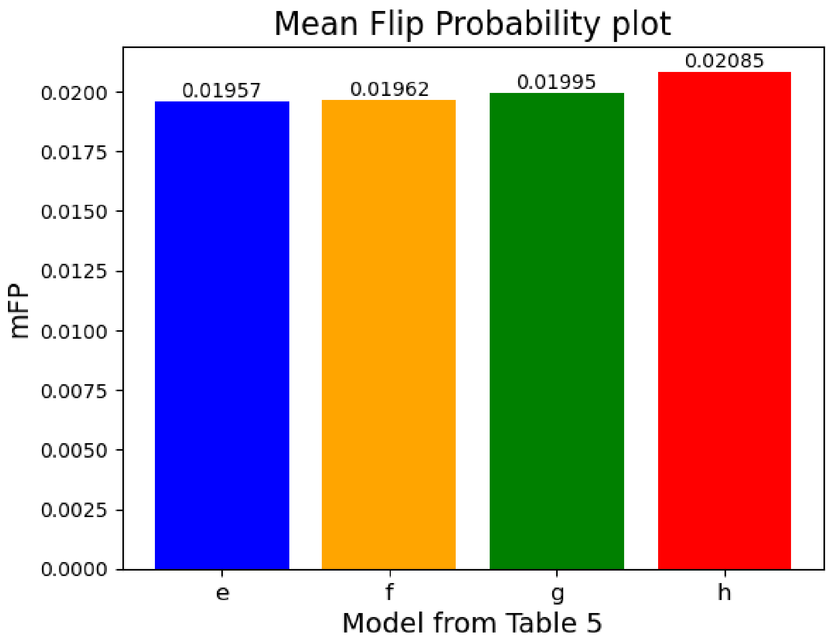

Figure 3: mFP from data in Table 5

3 Conclusion

In this work model performance against change in data distribution was conducted to train the Resnet18 and ConvNext-Tiny network with CIFAR-10 dataset that has been augmented using AugMix algorithm[5] and test on CIFAR-10, CIFAR-10C and CIFAR-10P. Non Pretrained Resnet18 model performed best with SGD and LambdaLR scheduler with Test error of 11.01 % and mCE of 16.755%, and mFP of 0.02225 and Pretrained Convnext-Tiny model best performed with AdamW optimizer and CosineAnnealingLR scheduler with inital Learning rate of 0.0001 and weight decay of 0.00001, with Test error of 7.65 %, mCE 13.071 % and mFP of 0.01995, furthermore it was seen that lowering the learning rate and weight decay improved the model performance, therefore it is possible to achieve more accuracy with Convnext-Tiny model, which could be a topic of further work, also in the model training performed in this work above it was seen that model seem to achieve convergence at around 96 epoch.As seen in the figure 3the Test error and Mean corruption error(mCE) is decreasing as hyperparameter is changed, the perturbation in the dataset has either no effect or slightly lower value, this means the perturbation in dataset does not have significant effect on the model performance with change in hyperparameter.

Bibliography

[1] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "AugMix: A simple data processing method to improve robustness and uncertainty," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[2] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *Proceedings of the International Conference on Learning Representations*, 2019.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[4] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11966–11976, 2022.

[5] D. Hendrycks*, N. Mu*, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple method to improve robustness and uncertainty under data shift," in *International Conference on Learning Representations*, 2020.

[6] G. Philipp, D. X. Song, and J. G. Carbonell, "The exploding gradient problem demystified - definition, prevalence, impact, origin, tradeoffs, and solutions," *arXiv: Learning*, 2017.