COMP 551 Project 4

Yuyan Chen, Ing Tian, Zijun Zhao

December, 2021

Abstract

In this project, we are trying to reproduce the main claims in this paper, Activate or Not: Learning Customized Activation[1]. In this study[1], the researchers have proposed a family of activation functions, called `ACON-C`, that should bring improvements "on a wide range of applications." During our course of experiments, we have found that effects of the proposed activation functions, `ACON-C` and `meta-ACON`, are model-dependent and task-dependent. We have conducted experiments with many models, such as ShuffleNet, ResNet, VGG16, etc, on two datasets, CIFAR-10 and CIFAR-100. As we have demonstrated, `ACON-C` and `meta-ACON` can bring positive, negative, and no impact depending on the experimental setting.

1 Introduction

The choice of activation function can greatly affect the performance of neural networks. Properties of activation functions, such as nonlinearity, range, and differentiability, can affect the number of active neurons in the neural networks. For this project, we aim to reproduce the result of the paper, Activate or Not: Learning Customized Activation [1]. We conducted experiments with models used in the paper, namely, ShuffleNet and ResNet, and also used VGG16 and AlexNet to test their claim that `ACON-C` and `meta-ACON`, two activation functions proposed by the authors, can improve the performance of neural networks. We found that the effectiveness of `ACON-C` and `meta-ACON` is task-dependent and model-dependent. The choice of data set, the number of parameters in the model, and the architecture of the neural network all affect the performance of `ACON-C` and `meta-ACON`.

2 Summary

The authors introduced a new family of activation function, `ACON`, that learns to activate the neurons or not. They defined `ACON-C` as

$$\begin{aligned} f_{ACON-C}(x) &= S_\beta(p_1x, p_2x) \\ &= (p_1 - p_2)x \cdot \sigma[\beta(p_1 - p_2)x] + p_2x, \end{aligned}$$

where β, p_1, and p_2 are learnable parameters. $f_{ACON-C}(x)$ is continuously differentiable, and its first derivative's bounds are learnable via parameter p_1, p_2. In contrast, other activation functions, such as ReLU and Swish, have a fixed upper and lower bound. Adjustable upper and lower bounds and differentiability are desired properties for an acivation function [2]. The linearity of the function is determined by β:

$$f_{ACON-C}(x) \xrightarrow{\beta \to \infty} max(p_1x, p_2x)$$
$$f_{ACON-C}(x) \xrightarrow{\beta \to 0} mean(p_1x, p_2x).$$

Hence, by switching β, `ACON` enables each neuron to adaptively activate or not. This property, according to the authors, is the key to improving the performance of neural networks, as it "improves generalization and transfers performance." They also introduced `meta-ACON` that learns β explicitly conditioned on the input sample via a neural network. The architecture of the neural network could be "layer-wise, channel-wise, or pixel-wise," and they implemented a channel-wise neural network for their experiments. The authors have conducted various experiments to support their claims and shown the effectiveness of `ACON` and `meta-ACON`.

3 Scope of Reproducibility

In this project, we aim to reproduce their claim that

- `ACON-C` and `meta-ACON` improve the performance of light neural networks significantly,
- `ACON-C` and `meta-ACON` are also effective for highly optimized large models,
- `ACON` activation function is robust and effective for a wide range of applications.

4 Methodology

The authors posted their codes of ShuffleNet, ResNet, `ACON-C`, and `meta-ACON`, so we can easily adapt their codes for our experiments.

However, we cannot conduct our experiments on ImageNet as they did due to the limit of computational power. Furthermore, it is impossible to confirm their results with provided pre-trained models as they do not specify the test images they used to calculate the accuracy.

Since they have claimed `ACON-C` and `meta-ACON` to be effective in a "wide range of applications," experiment results with other models and datasets should

also provide valid evidence that supports or rejects the claim.

4.1 Model description

We explored the models discussed in the paper, namely ShuffleNet0.5 and ResNet18 [1]. We also experimented with AlexNet and VGG16.

4.2 Datasets

CIFAR-100 The models in the paper were trained with ImageNet. Given the computational limit, we used CIFAR-100 instead. CIFAR-100 contains 100 classes, and each class consists of 500 training and 100 test images. The original size of all images is 32 by 32. We split the training set of CIFAR-100 into 45,000 training and 5,000 validation images. Since AlexNet, ResNet, and ShuffleNet do not apply to small images, we resized the images to 128 by 128 and subsequently normalized these images. VGG16 models were trained with normalized 32 by 32 images.

CIFAR-10 To explore whether the choice of the dataset can affect the performance of `ACON-C` and `meta-ACON`, we trained three variants of AlexNet with CIFAR-10. The images in CIFAR-100 and CIFAR-10 are identical, while the task is simpler with merely 10 labels. We used the same train-validation split for CIFAR-10. All images were resized and normalized as before.

4.3 Experimental setup

We have derived three variants for each model, denoted as `ReLU`, `ACON`, and `META`. Models labeled `ReLU` have ReLU as their activation functions; models labeled `ACON` have their ReLU functions replaced by `ACON-C` in convolutional layers; similarly, models labeled `META` have their ReLU functions replaced by `meta-ACON` in convolutional layers.

For VGG16, we explored two versions of VGG16, namely VGG16-3 and VGG16-6. For VGG16-3, we removed the last three convolutional layers from VGG16. Likewise, we removed the last six convolutional layers from VGG16 to obtain VGG16-6. We had three versions of AlexNet: the origunal AlexNet, AlexNet-s1, and AlexNet-s2. AlexNet-s1 and AlexNet-s2 were obtained by reducing all channel sizes to $\frac{3}{4}$ or $\frac{1}{2}$ of the original channel size respectively.

For ShuffleNet0.5 and ResNet18, we adapted their codes and trained them on CIFAR-100.

Due to various convergence speeds, models were trained for different numbers of epochs. AlexNets and ResNet were trained for 50 epochs, whereas ShuffleNet and VGG16s were trained for 100 epochs.

5 Results

As to get the test accuracy for each model, we pick the snapshot of the model where its validation accuracy is within 0.001 of the best validation accuracy and is trained with the fewest number of epochs. For each experimental setup, we have recorded the model's highest validation accuracy and test accuracy on CIFAR-100 and CIFAR-10, as in Table 1 and Table 2.

ResNet18 The experiment results for variants of ResNet18 on CIFAR-100 are depicted in Figure 1. Three variants of ResNet18 have almost identical performances. The choice of the activation function did not affect the convergence speed and the accuracy of ResNet18.

ShuffleNet0.5 Results from ShuffleNet0.5 (Figure 2) show that `meta-ACON` improved the performance of ShuffleNet0.5 significantly, but `ACON-C` has no effect on the performance of the model. In addition, both `ACON-C` and `meta-ACON` increase the convergence speed of ShuffleNet0.5.

AlexNet For AlexNet (Figure 3, Figure 8), AlexNet-s1 (Figure 4, Figure 9), and AlexNet-s2 (Figure 5, Figure 10), we have observed that using `ACON-C` and `meta-ACON` can significantly improve the accuracy and convergence speed. Also, for all three settings, the improvements of `ACON-C` and `meta-ACON` from `ReLU` are identical.

VGG16 For VGG16-3 (Figure 6), both training and validation accuracy of the `ReLU` variant are consistently higher than two other variants. However, all three variants have similar performance for VGG16-6 (Figure 7).

6 Discussion

Given the results, we have observed that the impacts of `ACON-C` and `meta-ACON` are model-dependent and task-dependent.

6.1 Model

Complexity In the paper, they found the improvement on light neural networks was greater than on deep neural networks, and whether the model is light or great is judged by its number parameters. Besides, they had different experiment designs for light and deep neural networks to prevent overfitting. However, they did not provide a clear threshold in terms of the number of parameters for "light" and "deep" models.

Intra-model comparsion According to the paper, `ACON-C` and `meta-ACON` improve the performance

more on light models than on deep models. The authors also mentioned that changing all ReLUs into meta-ACON might cause overfitting. Hence, we propose the hypothesis that reducing the number of parameters can increase the improvement brought by ACON-C and meta-ACON

As to VGG16-3, the introduction of ACON-C and meta-ACON brings negative impacts. When we reduced VGG16-3 to VGG16-6, the number of parameters is reduced to 1.74M, and using ACON-C and meta-ACON does not have a negative impact on the performance, which corresponds to our hypothesis.

In contrast to VGG16, we found that the more parameters the AlexNet has, the greater the increase in accuracy for the ACON and META variants. However, ACON-C and meta-ACON have an almost identical effect in all three settings, namely AlexNet, AlexNet-s1, and AlexNet-s2.

At this point, it is obvious that the relation between model complexity and performance of VGG16 is different from that of AlexNet. Hence, we conclude that the relation between the effectiveness of ACON-C and meta-ACON and the model complexity is model-dependent.

Neural network architecture VGG16 and AlexNet have a very different architecture from MobileNet, ShuffleNet, and ResNet. Also, VGG16 and AlexNet require activation functions between fully connected layers, while the other three do not. The difference in the architecture might be the reason why the results of AlexNet and VGG16 are not comparable to the results shown in their paper. Moreover, the introduction of meta-ACON and ACON-C can bring positive impacts, no impacts, and negative impacts depending on the model, and it is impossible to determine exactly when the ACON activation functions should be beneficial.

6.2 Data set

The results of ShuffleNet0.5 and ResNet18 of our experiments are different from those in their experiments. We propose the following hypothesis: the performance of ACON-C and meta-ACON is task-dependent. They are more suitable for complex tasks while using these activation functions for simpler tasks might not work due to overfitting.

CIFAR-100 and ImageNet The authors claimed that meta-ACON improved the accuracy more than ACON-C, and both functions had a positive impact on the performance of models over ReLU. However, we found that the choice of activation function had little impact on the performance of both ShuffleNet and ResNet on CIFAR-100. Compared with using ReLU, the test accuracy increased by 0.67% with meta-ACON, but decreased by 0.05% with ACON-C. For ResNet18, the test accuracy decreased by 1.44% and 2.3% respectively for meta-ACON and ACON-C. One possible explanation of why we cannot reproduce the effects listed in the paper is the difference in datasets.

CIFAR-100 and CIFAR-10 To further test our hypothesis, we trained AlexNet, AlexNet-s1, and AlexNet-s2 on CIFAR-10. For ACON-C, the increase in test accuracy of AlexNet, AlexNet-s1, and AlexNet-s2 is 3.74%, 3.86%, and 2.82% respectively on CIFAR-10 compared to ReLU, while 14.17%, 9.55%, and 7.99% on CIFAR-100 compared with ReLU. For meta-ACON, the increase in test accuracy of AlexNet, AlexNet-s1, and AlexNet-s2 is 3.97%, 3.60%, and 2.71% respectively on CIFAR-10 over ReLU, while 12.63%, 8.79%, and 8.17% on CIFAR-100 over ReLU. Hence, the results has confirmed our hypothesis as ACON-C and meta-ACON can bring more improvements on more complicated tasks.

Based on the experiment results, we conclude that the effectiveness of ACON-C and meta-ACON depends on the complexity of the classification task. Replacing ReLU with ACON-C or meta-ACON leads to a greater increase in the test accuracy for a more complicated classification task.

7 Conclusion

As we reproduce the results in this study[1], we have evidence that partially supports the study's claims. Our experiments with AlexNets and ShuffleNet0.5 support the authors' claim that ACON-C and meta-ACON can improve the performance of neural networks. However, ACON-C and meta-ACON are not guaranteed to improve the performance of models, whether the model is small or large, rather, the effectiveness of these two activation functions is model-dependent and task-dependent.

8 Statement of contribution

Yuyan Chen - Report Writeup
Zijun Zhao - Code Implementation
Ing Tian - Experiment Design

References

[1] Ma, Ningning, et al. "Activate or Not: Learning Customized Activation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[2] Snyman, Jan A. Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. Springer, 2005.

Appendix

Table 1: Validation and training accuracy of each model on CIFAR-100

Model	Test	Validation
ResNet18 ReLU	**0.5685**	**0.5713**
ResNet18 ACON	0.5455	0.5529
ResNet18 META	0.5541	0.5678
ShuffleNet0.5 ReLU	0.3862	0.3953
ShuffleNet0.5 ACON	0.3821	0.3953
ShuffleNet0.5 META	**0.3929**	**0.4273**
AlexNet ReLU	0.4339	0.4843
AlexNet ACON	**0.5756**	**0.5710**
AlexNet META	0.5602	0.5678
AlexNet-s1 ReLU	0.4496	0.4384
AlexNet-s1 ACON	**0.5451**	0.5425
AlexNet-s1 META	0.5375	**0.5678**
AlexNet-s2 ReLU	0.4525	0.4590
AlexNet-s2 ACON	0.5324	0.5363
AlexNet-s2 META	**0.5342**	**0.5365**
VGG16-3 ReLU	0.4583	**0.6073**
VGG16-3 ACON	**0.5870**	0.5735
VGG16-3 META	0.5842	0.5692
VGG16-6 ReLU	0.6209	0.6073
VGG16-6 ACON	**0.6376**	**0.6198**
VGG16-6 META	0.6350	0.6167

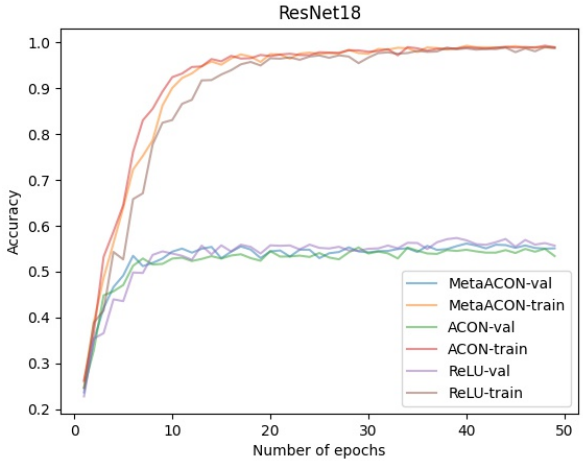

Figure 1: Training and validation accuracy of ResNet18 on CIFAR-100

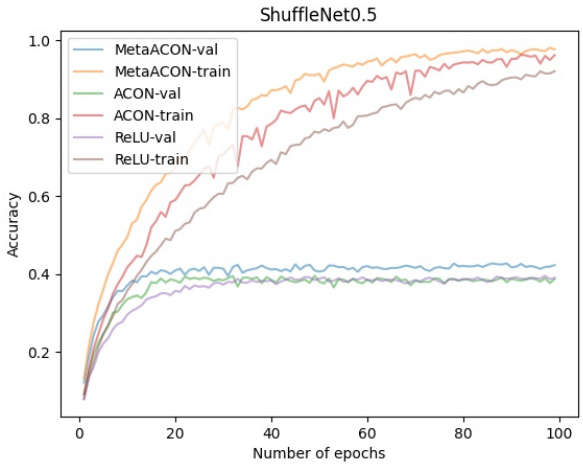

Figure 2: Training and validation accuracy of ShuffleNet0.5 on CIFAR-100

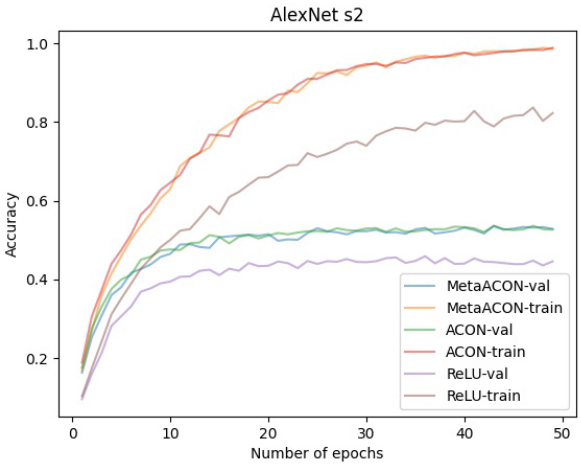

Figure 5: Training and validation accuracy of AlexNet-s2 on CIFAR-100

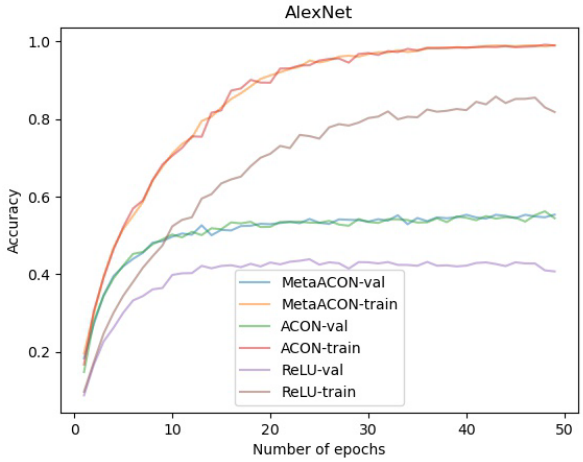

Figure 3: Training and validation accuracy of AlexNet on CIFAR-100

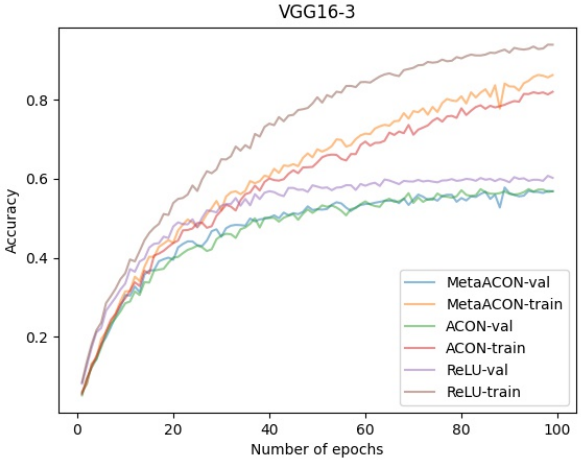

Figure 6: Training and validation accuracy of VGG16-3 on CIFAR-100

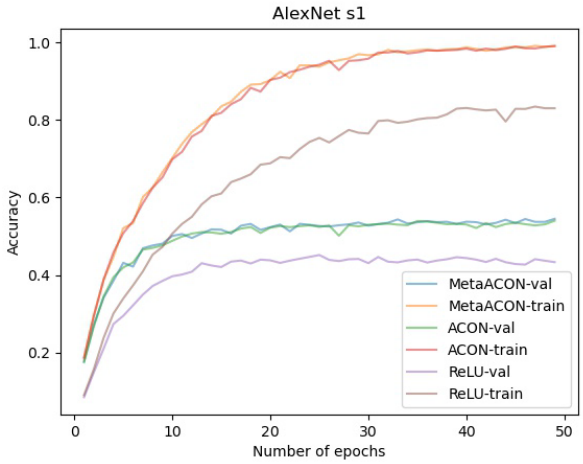

Figure 4: Training and validation accuracy of AlexNet-s1 on CIFAR-100

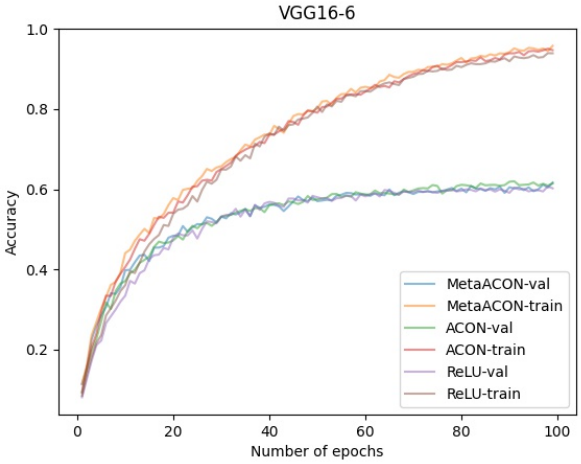

Figure 7: Training and validation accuracy of VGG16-6 on CIFAR-100

Table 2: Validation and training accuracy of AlexNets on CIFAR-10

Model	Test	Validation
AlexNet ReLU	0.7901	0.7986
AlexNet ACON	0.8275	**0.8449**
AlexNet META	**0.8298**	0.8384
AlexNet-s1 ReLU	0.7928	0.8027
AlexNet-s1 ACON	**0.8314**	**0.8416**
AlexNet-s1 META	0.8288	0.8384
AlexNet-s2 ReLU	0.7914	0.7953
AlexNet-s2 ACON	**0.8196**	0.8288
AlexNet-s2 META	0.8185	**0.8339**

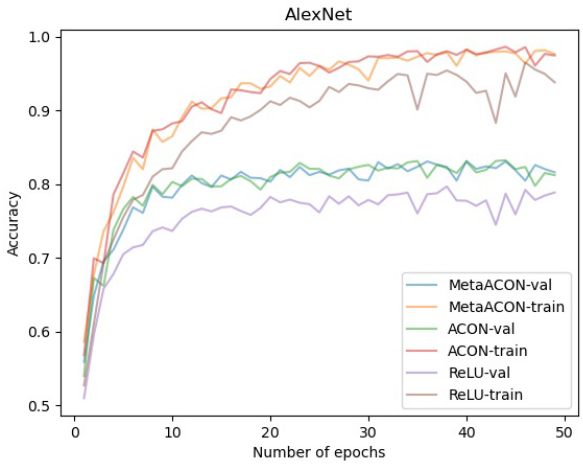

Figure 8: Training and validation accuracy of AlexNet on CIFAR-10

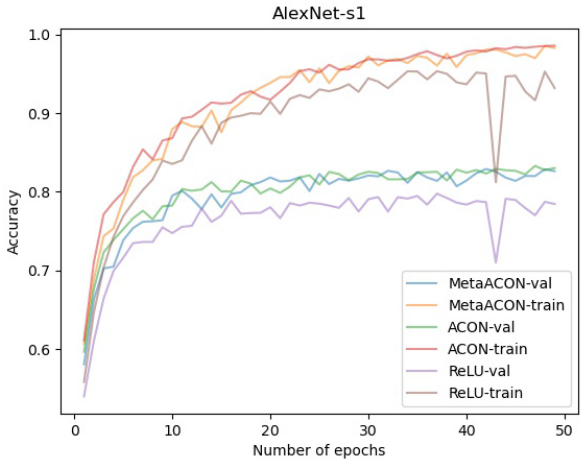

Figure 9: Training and validation accuracy of AlexNet-s1 on CIFAR-10

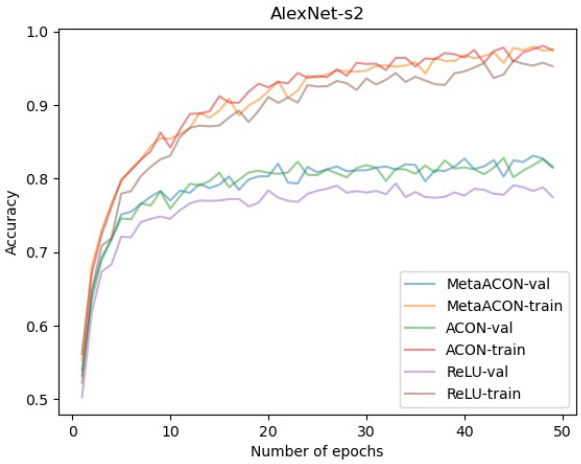

Figure 10: Training and validation accuracy of AlexNet-s2 on CIFAR-10