Reproducibility report formatting instructions for ML Reproducibility Challenge 2020

Yao Chen
yao.chen5@mail.mcgill.ca
260910483

Chuyang Zhang
chuyang.zhang@mail.mcgill.ca
260840707

Hanzhi Zhang
hanzhi.zhang@mail.mcgill.ca
260908416

Reproducibility Summary

Abstract

In this mini-project, our group reproduced the results from the "*Image Classification Algorithm Based on Improved AlexNet*" paper. We then modified the model by deleting two convolutional layers to evaluate the importance of the various model components. Furthermore, we tried to change the hyperparameters and increase the prediction accuracy. Since the ImageNet dataset is too large to use, we used the Cifar-10 dataset instead with the same AlexNet architecture. We found that the AlexNet architecture would significantly overfit when using Cifar-10 as input, which would have a training accuracy of 100% using the parameters from the AlexNet paper.

1. Introduction

AlexNet is a large, deep convolutional neural network (CNN) and can classify 1.2 million high-resolution images in the ImageNet dataset into 1000 different classes. After winning first place in the ImageNet LSVRC-2010 contest, it became famous. The AlexNet model has more than 10% higher accuracy than the second place. The AlexNet consists of eight layers: the first five are the convolutional layers, then three fully connected layers. Due to the time limit, we reproduced the original AlexNet architecture and a modified AlexNet architecture with deconvolutional layers and tested the accuracy on the Cifar-10 dataset, which was significantly smaller than the ImageNet dataset. Surprisingly, most of the results reproduced were not the same as stated in the paper.

2. Scope of Reproducibility

Since the AlexNet architecture can classify millions of images with thousands of classes, it may be too big for the Cifar-10 datasets with 60,000 images and 10 different classes. Li's group developed a modified AlexNet, named new_AlexNet. It contained four convolutional layers, two deconvolutional layers (transposed convolutional layers), and two fully connected layers. They trained the Cifar-10 datasets on both the original AlexNet and the new_AlexNet.

In this project, we aim to reproduce the main experiments in Li's paper as follows:

- Implementing the AlexNet and new_AlexNet architecture using the TensorFlow package obtained different parameters and models from the paper.
- Tuning the value of the "dropout" on the original AlexNet increases the test accuracy and prevents overfitting.
- Obtained 40% less test accuracy on the new_AlexNet and 13% less on the AlexNet model with the Cifar-10 dataset.

3. Methodology

Since we couldn't find the author's code corresponding to the paper, we aim to re-implement the approach stated in the article. We obtained most of the original AlexNet code from the book *Computer Vision Using Deep Learning* with minor modifications [1] The modified AlexNet code was implemented by ourselves using the network structure detailed parameters in the *Image Classification Algorithm Based on Improved AlexNet* paper by Shaojuan Li [2]. We implemented the code in the Google Colab and used GPU acceleration during the runtime when training the model.

3.1 Model descriptions

Two models are used in this project, the AlexNet and the new_AlexNet, as described above.
The detailed parameters for both models are shown in Appendix Tables I and II.

In total, the original AlexNet had 21,622,154 trainable parameters whereas the modified AlexNet had 844,938 trainable parameters. Notice that the new_AlexNet had a *CropCenter* layer right after receiving the input to make the shape 28 by 28 pixels instead of 32 to be consistent with Li's paper, but the input figures are still 32 by 32 pixels. None of these two models was pre-trained.

3.2 Dataset

The Cifar-10 dataset consists of 60,000 32 by 32 pixels colour images in 10 classes, with 6000 images per class[3]. Among these 60,000 images, 50,000 are used as the training data and 10,000 as the test data. We pre-processed the images first by subtracting the mean and dividing the standard deviation. We then used *ImageDataGenerator* from the *Keras* package to allow the horizontal flip of the images and width or height shift by 0.115.

Finally, use this link to download the data in python.
You can also import the data using the *Keras* datasets package. The code to import is demonstrated below:

```
from keras.datasets import cifar10
(x_train, y_train), (x_test, y_test) = cifar10.load_data()
```

3.3 Hyperparameters

In Li's paper, since only a few hyperparameters were specified (batch size and the number of cycles), the unspecified hyperparameters were used from the original AlexNet paper. In the process of training, each batch size is 128, a total of 120 cycles of training, the number of iterations is 390, the initial learning rate is 0.01, the loss is categorical cross-entropy loss, and the momentum is 0.9. We manually searched the dropout rate of the dropout containing layers with the values 0.25, 0.5, and 0.9 in the original AlexNet model. At the dropout equals 0.9, we found that the prediction had the highest accuracy (72%) and closer training/ validation loss (the least overfitting model).

3.4 Experimental Setup and Code

As described above, the cifar10 data were imported from the *Keras* datasets, preprocessed and fed into the model with specific parameters. We then train the data using the provided 50,000 training data and test the accuracy using the 10,000 testing data for both models. After tuning the hyperparameter for the original AlexNet, we found that the accuracy could reach 72.2%. The accuracy for the Cifar-10 dataset using the modified AlexNet could reach 50%.
The code for this project can be found via this link and this link.

3.5 Computational Requirements

The full implementation was done on the Google Colab Pro cloud computing. The CPU used was n1-highmem-2 instance 2vCPU @ 2.2GHz, and the GPU used was NVIDIA Tesla K80. With the GPU acceleration, the average running time for each batch is around 23 seconds, and the average time for training the model with 120 batches would be around 60 minutes for both AlexNet and new_AlexNet. This model could also be trained without GPU acceleration with a much longer waiting time. The average running time for each batch would be around 8 minutes without GPU.

4 Results

The highest test accuracy for the original AlexNet was 72.2%, compared to 84.2% in the paper. The test accuracy for the new_AlexNet only reached 57.2%, lower than the original AlexNet, and also lower than 87.2% reported in the paper. The loss and accuracy plot for the new_AlexNet was interesting and unexpecting.

4.1 Results Reproducing the original paper

We implemented both AlexNet and new_AlexNet architectures. Besides, we used these two models on the Cifar-10 dataset and obtained the loss and accuracy for the training. We compared the parameters and the test accuracy with the original paper and reported in the 4.1.1 and the 4.1.2 sections.

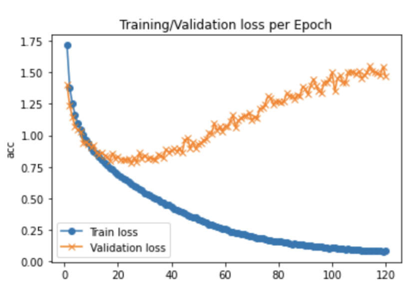
4.1.1 Model implementations and parameters

Due to the lack of the source code corresponding to the article, our group implemented the AlexNet and new_AlexNet models based on their description in the paper. The comparison between the model parameters from the reference paper and our implementation is shown in Appendix Table III.

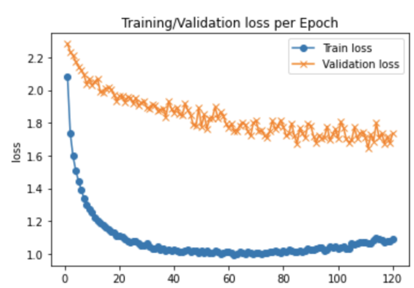
Table III shows that without the source code or a more accurate description of the AlexNet model, it is doubtful to reproduce the exact model as in the paper.

4.1.2 Model behaviour on Cifar-10 dataset

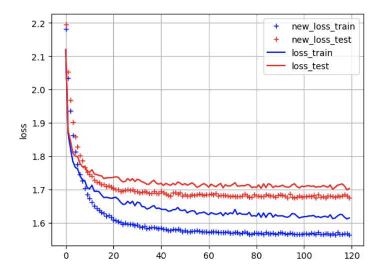
We then trained both models and generated the loss and accuracy with respect to the number of epochs plots and compared them with the plots in the reference paper. The figures are shown below in Figure 1.



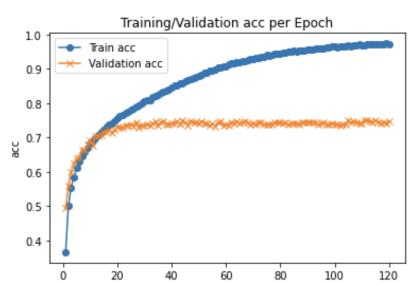
A)



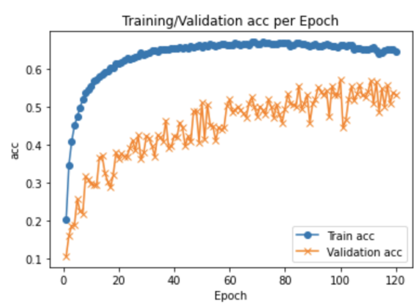
B)



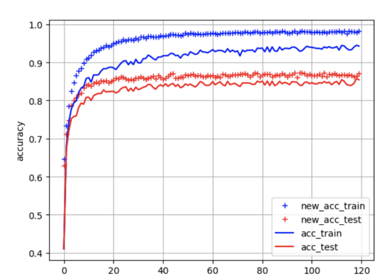
C)



D)



E)



F)

Figure 1: Comparison of the accuracy and loss between the paper and our experiments. There are 120 epochs, and the batch sizes are 128 for all experiments. A) The loss per epoch for the original AlexNet in our implementation. B) The loss for the new_AlexNet in our implementation. C) The loss for both implementations in the reference paper. D) The accuracy of the original AlexNet in our implementation. E) The accuracy of the new_AlexNet in our implementation. F) The accuracy of both implementations in the reference paper.

We can see from the figure that our test accuracy reached 70% for the AlexNet and only 50% for the new_AlexNet without overfitting. The accuracies reported in the literature are 83.4% for AlexNet and 87.4% for the new_AlexNet. The difference is significant. Besides, our model had more problems with overfitting than the model in the literature. This is mainly discussed in Section 5.

4.2 Result Beyond the original paper

We tuned the optimal batch size and the dropout value for the original AlexNet architecture on the Cifar-10 dataset.

4.2.1 Dropout tuning

To test how the dropout value affects the final test accuracy, we manually selected three dropout values: 0.25, 0.5, and 0.9 for the dropout layers in AlexNet architecture. The training and validation accuracy is shown below in Figure 2.

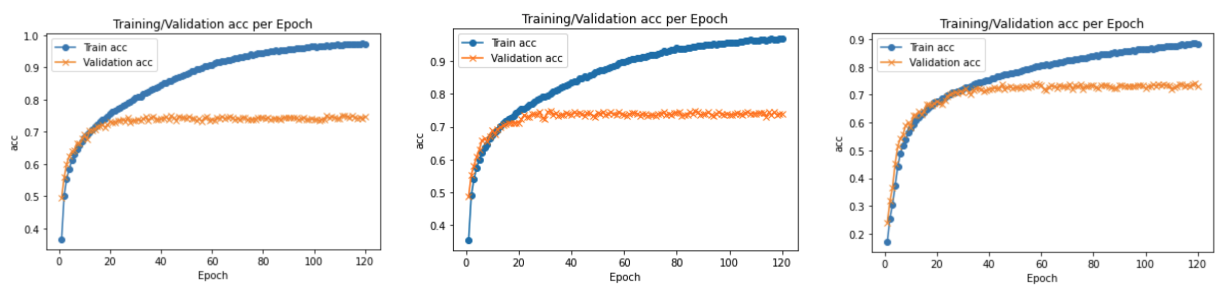


Figure 2: Training and validation accuracy with different dropout values. Left: the dropout is 0.25, Middle: the dropout is 0.5, and Right: the dropout is 0.9.

The test accuracy was 70%, 71.2%, and 72.2%, respectively. We can also see from the figures that the model tends to overfit less when the dropout value increases.

5 Discussion

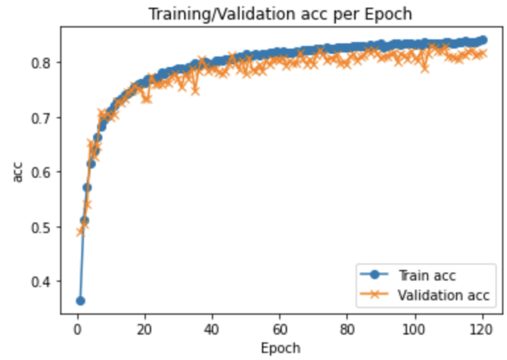
5.1 What was easy

The paper's logic was easy because they simply modified the original AlexNet architecture to make the model fits better for the smaller datasets. The results were also clearly stated, and the math doesn't require advanced knowledge of calculus to follow.

5.2 What was difficult

Since the literature didn't provide the source code, we have to implement the code to perform the experiments by ourselves. We build our model based on the network structure with detailed parameters in the paper. After setting up all the values of hyperparameters identical to the paper and training the model, we found our model accuracy distinct from the paper's accuracy. The test accuracy for the original AlexNet was 72.2%, compared to 84.2% in the paper. The test accuracy for the new_AlexNet only reached 57.2%, lower than the 87.2% reported in the paper, which is 30% lower. We found both our original and new_AlexNet model exist overfitting issues under the same hyperparameter setting as the paper. We then tried pruning hyperparameters by ourselves, and we didn't find an obvious improvement in leading the result close to the result given by the paper. Without providing the source code, it is for sure very difficult to reproduce the exact result as the original paper.

However, we then restructured our new_AlexNet model aside based on the structure given in the paper — two Convolution layers were dropped in order to resolve the overfitting problem. Meanwhile, to avoid the underfitting, we then adjust the dropout rate between each layer to either 0.3 or 0.5. Referring to the graph at the right, our new_AlexNet model finally reached the test accuracy of 81%, compared with 87.2% reported in the paper.



5.3 Communication with original authors

Due to the time limit, our group didn't communicate with the original authors.

Statements of contribution

The group members distributed the workload equally.
Yao: graph generation, hyperparameter tuning, creativity;
Chuyang: write-up, graph generation, hyperparameter tuning, creativity;
Hanzhi: write-up, creativity.

Reference

[1] V. Verdhan, *Computer Vision using deep learning: Neural network architectures with python and keras*. New York, NY: Apress, 2021.

[2] S. Li, L. Wang, J. Li, and Y. Yao, "Image Classification Algorithm Based on Improved AlexNet," Journal of Physics: Conference Series, 01-Feb-2021. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1813/1/012051. [Accessed: 24-Apr-2022].

[3]A. Krizhevsky, "The CIFAR-10 dataset," *CIFAR-10 and CIFAR-100 datasets*. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html. [Accessed: 24-Apr-2022].

Appendix

Table I: The original AlexNet Network Structure detailed parameters

Network Layer	Output Shape	Kernel Size/ Strides	Activation Function	Dropout	Pooling Layer/ Strides	Num Parameters
Input	32, 32, 3					
Conv1	4, 4, 96	11, 11/ 4, 4	ReLU	No	3, 3/ 2, 2	34944
Conv2	2, 2, 256	5, 5/ 1, 1		No	3, 3/ 2, 2	614656
Conv3	2, 2, 384	3, 3/ 1, 1		No	No	885120
Conv4	2, 2, 384	3, 3/ 1, 1		No	No	1327488
Conv5	1, 1, 256	3, 3/ 1, 1		No	3, 3/ 2, 2	884992
FC1	4096	N/A		0.5	N/A	1052672
FC2	4096	N/A		0.5	N/A	1052672
FC3	**10**	N/A	Softmax	No	N/A	40970

Table II: The modified new_AlexNet Network Structure detailed parameters

Network Layer	Output Shape	Kernel Size/ Strides	Activation Function	Dropout	Pooling Layer/ Strides	Num Parameters
Input	28, 28, 3					
Conv1	28, 28, 64	3, 3/ 1, 1	ReLU	No	No	1792
Conv2	14, 14, 64	3, 3/ 1, 1		0.8	3, 3/ 2, 2	36928
Conv3	14, 14, 128	3, 3/ 1, 1		No	No	73856
Conv4	7, 7, 256	3, 3/ 1, 1		0.8	3, 3/ 2, 2	295168
DeC1	4, 4, 128	3, 3/ 1, 1		0.5	3, 3/ 2, 2	295040
DeC2	2, 2, 64	3, 3/ 1, 1		0.5	3, 3/ 2, 2	73792
FC1	256	N/A		0.5	N/A	65792
FC2	10	N/A	Softmax	No	N/A	2570

Table III: Comparison of parameters between the reference paper and our work

	Model	Num Parameters	Fully Connected Parameters %
Reference	AlexNet	8,819k	95.4%
Our	AlexNet	21,622k	83%
Reference	new_AlexNet	586k	11.6%
Our	new_AlexNet	884k	8.1%

Table IV: The restructured new_AlexNet Network Structure detailed parameters

Network Layer	Output Shape	Kernel Size/ Strides	Activation Function	Dropout	Pooling Layer/ Strides	Num Parameters
Input	28, 28, 3					
Conv1	28, 28, 64	3, 3/ 1, 1	ReLU	No	No	1792
Conv2	14, 14, 64	3, 3/ 1, 1		0.5	3, 3/ 2, 2	73856
DeC1	4, 4, 128	3, 3/ 1, 1		0.3	3, 3/ 2, 2	147584
DeC2	2, 2, 64	3, 3/ 1, 1		0.3	3, 3/ 2, 2	73792
FC1	256	N/A		0.3	N/A	65792
FC2	10	N/A	Softmax	No	N/A	2570