

Deep Learning HW3

曾以諾

Institute of Data Science
National Cheng Kung University
Tainan, R.O.C.
re6111032@gs.ncku.edu.tw

Abstract—This report is an description document for HW3. And the GitHub link is as follows

GitHub Link:

https://github.com/butterfly2012010/DeepLearning_HW3

I. INTRODUCTION

In homework 3, there are three requirements to complete,

- Implement LeNet5 (any version) by Tensorflow (Keras is ok) and Pytorch.
- Performance Comparison (accuracy, inference time, space complexity, and flops) of the TF/Pytorch and your handcrafted versions.
- In the TensorFlow version, please make a "static graph" LeNet5 and compare it with a "dynamic" one in inference and training time.

Incidentally, there is an extra bonus in this homework requirement, and I do the network pruning with keras API.

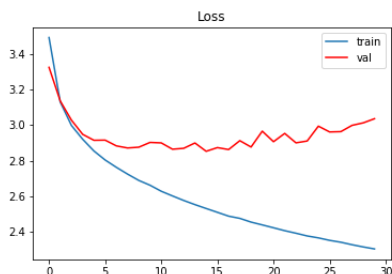
Model compression: Apply one of the model compression techniques like quantization, pruning, and matrix factorization. Evaluate the performance and speed (run-time) for the compressed and naive models.

II. IMPLEMENTATION

A. LeNet5 by Tensorflow Keras

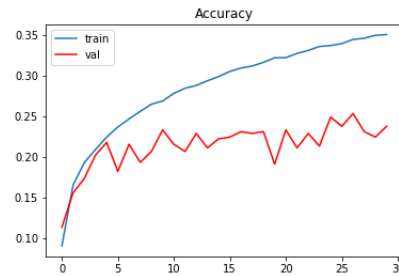
1) Loss

The plot of train and validation loss



2) Accuracy

The plot of train and validation accuracy



3) predicted results

- On validation set, the (top-1) accuracy is 21.33 %.
- On testing set, the accuracy is 22.22 %.

4) static graph

- training time: 5.5222 seconds
- inference time: 0.2327 seconds

5) dynamic graph

- training time: 3.1486 seconds
- inference time: 0.0718 seconds

6) Extra Bonus: Network Pruning

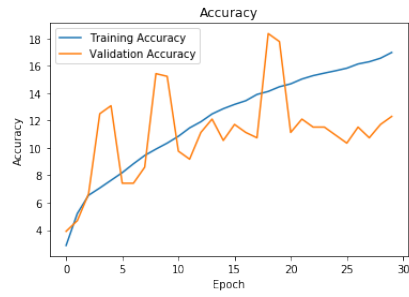
- Test loss: 2.9053
- Test accuracy: 22.44 %

B. LeNet5 by Pytorch

1) Loss



2) Accuracy



3) predicted results

- On validation set, the accuracy is 14 %.
- On testing set, the accuracy is 16 %.

C. Performance Comparison

	LeNet5 by Tensorflow	LeNet5 by PyTorch	LeNet5 by handcraft
accuracy (%)	22.22	16.00	20.00
inference time (seconds)	0.0718	0.0022	40.1202
space complexity	0.2495 MB on cuda	4.4888 MB on cuda	0.2495 MB
(number of parameters)	(65406)	(65406)	(65406)
flops	0.0877 MFLOPs	167.738624 MFLOPs	-

III. REFERENCES

REFERENCES

- [1] ML_From_Scratch, <https://github.com/eriklindernoren/ML-From-Scratch>
- [2] ChatGPT