

Deep Learning HW3 - LeNet/Computational Graph

R76111206 曾凱揚
May 28, 2023

Abstract - This report is an description document for previous Homework. And the GitHub link is as follow

*Github Link, folder HW3
<https://github.com/vangtseng219/DeepLearning>*

I. INTRODUCTION

本次作業將延續作業二所使用到圖片萃取特徵與結果透過實作進行圖片的分類任務，將圖片數據套用至各個模型訓練，最終比較各類模型的成效。

本次作業需使用Pytorch與Tensorflow實作，其模型的架構為ChatGPT所生成，由於使用套件來實作，也有遇到維度的問題，也是透過ChatGPT詢問獲得解答。

此次作業也須計算inference time、space complexity以及FLOPs，另外tensorflow版本的需計算動態圖與靜態圖的時間差異，2.0主要使用的是動態圖，然而通過@tf.function裝飾器，可以將動態圖轉換成靜態計算圖。

II. METHOD

本作業皆使用 LeNet5 並且分別採用 Handcraft、PyTorch與Tensorflow實作，

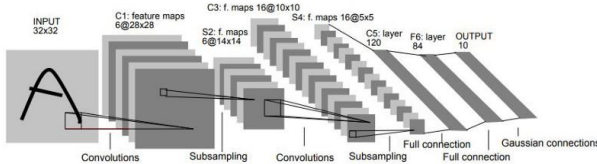
A. Model

為了進行實驗結果的比較，三種版本皆採用相同的 Hyperparameter

1) Hyperparameter

- a) Batch size: 32
- b) Epoch: 10
- c) Learning rate: 0.001

2) Model



B. Lenet5 Version

1) Handcraft

使用 Github 上由 chuanqi305 所提出的開原程式碼，並將作者原先使用 ReLU 激活函數更換成 Sigmoid 激活函數。

2) Tensorflow

使用 ChatGPT 生成 LeNet5 的架構，並將原始輸出的維度更改為 50，符合所使用的資料集，所使用的 2.X 版的 tensorflow，為動態計算圖，並使用 @tf.function 轉換成靜態計算圖。

3) PyTorch

使用 ChatGPT 生成 LeNet5 的架構，並將原始輸出的維度更改為 50，符合所使用的資料集。

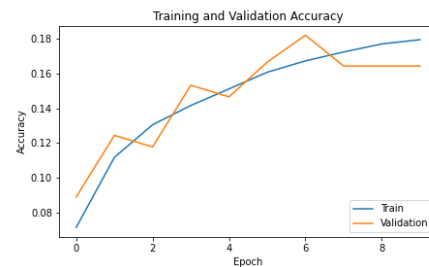
III. RESULT

A. Accuracy

1) Tensorflow

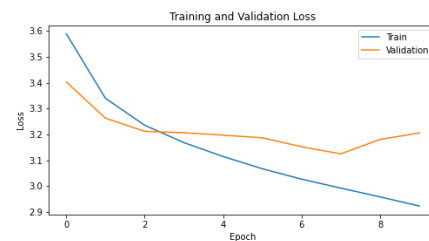


2) Pytorch

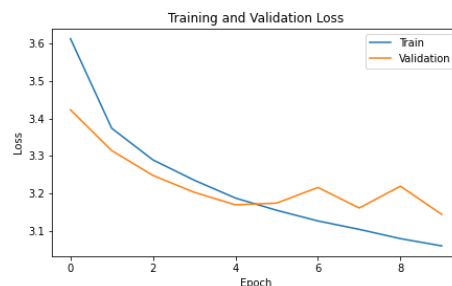


B. Loss

1) Tensorflow



2) Pytorch



C. predicted result

1) validation set

	Accuracy	Loss
Tensorflow	0.1644	3.2058
PyTorch	0.1644	3.1441

2) testing set

	Accuracy	Loss
Tensorflow	0.1956	3.0579
PyTorch	0.1933	3.0858

D. Compare

1) Accuracy

	Accuracy
Handcraft	0.0178
Tensorflow	0.1956
Pytorch	0.1933

使用套件的結果相對於手寫好上非常多

2) time

	Inference time / seconds	MFLOPs	Space Complexity / Million
Handcraft	0.192792	0.3738	0.0651
Tensorflow	0.005083	0.0084	0.0651
Pytorch	0.000497	13.4362	0.0651

E. Tensorflow - static graph vs dynamic graph

1) Comparison

	static graph	dynamic graph
Training time	1.1024 seconds	132.42 seconds
Inference time	0.001622 seconds	0.005083 seconds

IV. CONCLUSION

透過本次的作業了解到如何計算效能，並實作目前深度學習常見的架構Pytorch與Tensorflow，然而在實作本次作業中，可以發現兩者架構輸入的維度順序不同，需做處理才可以順利運作。

REFERENCES

- [1] LeNet5 - Handcraft
<https://github.com/chuanqi305/LeNet5>
- [2] ChatGPT
- [3] Tensorflow
<https://tianchi.aliyun.com/notebook/188638>
- [4] Profiler
<https://pypi.org/project/model-profiler/>