

# Assignment 2 - LeNet&Computational Graph \*

Xiang Yu-Ye

Mechanical Engineering  
National Cheng Kung University  
Tainan City East Dist. Daxue Rd., Taiwan  
N16114445@gs.ncku.edu.tw

**Abstract** -使用三種分類器Two Layer Net、LeNet5、修改LeNet5，來分類50個動物類別。

**Index Terms** – Deep Learning、LeNet5、Fully Connected、Convolution

## I. INTRODUCTION

本次作業目標為手刻Two Layer Net、LeNet5、修改LeNet5，使用python搭配numpy來完成，參考了老師給的Github連結 [1]，及自行上網搜尋的 [2]，所有code都放在 [https://github.com/Jason890102/Deep-Learning/tree/main/Assignment\\_2\\_LeNetComputational\\_Graph](https://github.com/Jason890102/Deep-Learning/tree/main/Assignment_2_LeNetComputational_Graph)。

## II. METHOD

### A. Two Layer Net

Two Layer Net架構圖如 Fig. 1，使用兩層 Fully Connected。Fig. 2.為training的Loss curve，以及train&Test的accrly，訓練50個epochs。

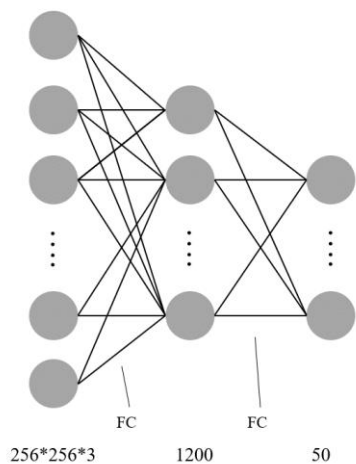


Fig. 1. Two Layer Net Architecture

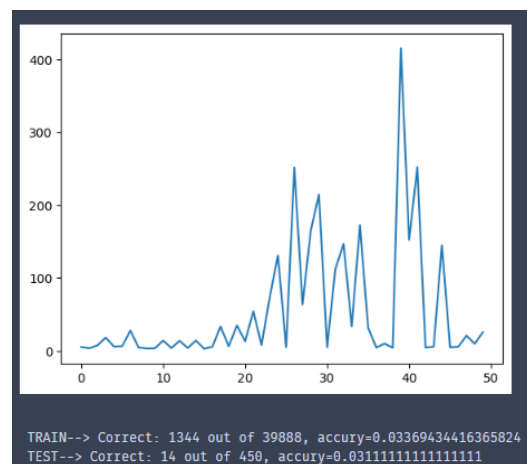


Fig. 2. Loss Curve、train&test accrly

### B. LeNet5

LeNet5的Net架構圖如Fig. 3，分為50個類別，值得注意的是將所有圖片resize成(256\*256)，Fig. 4. 為LeNet5 training的Loss curve，以及train&Test的accrly，訓練10個epochs。

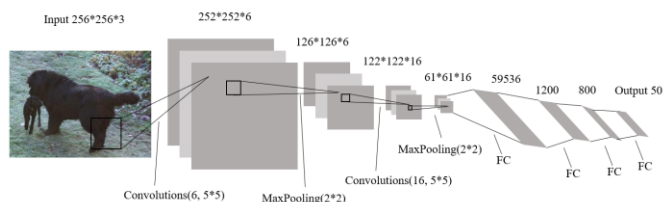


Fig. 3. LeNet5 Architecture

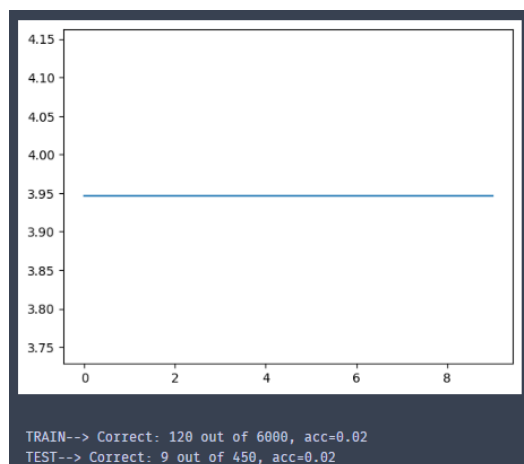


Fig. 4. LeNet5 Loss Curve、train&test accrly

### C. Improved LeNet5

Improved LeNet5的Net架構圖如Fig. 5.，分為50個類別，值得注意的是將所有圖片resize成(254\*254)，Fig. 6. 為LeNet5 trianing的Loss curve，以及trian&Test的accury，訓練10個epochs，可以看出accury稍微好一點，雖然一樣都很爛就是了。

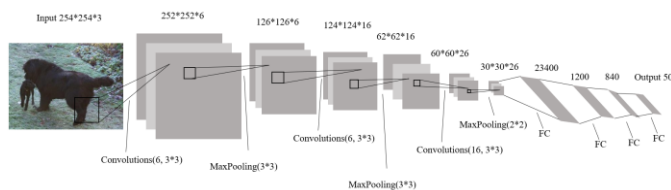


Fig. 5. Improved LeNet5 Architecture

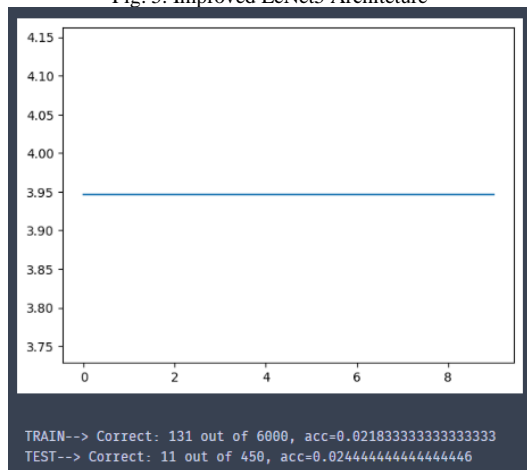


Fig. 6. Improved LeNet5 Loss Curve 、train&test accury

### III. CONCLUSION

在寫code的過程中，我是參考老師給的Github連結裡的code，有先用自己所知道的知識來審視每一行code在做甚麼運算，看完後想說先run一次看看可不可以跑得起來，沒想到還跑不起來，不斷的出現error，在debug期間，注意到了許多的細節，例如FC是使用He來做Weight initialization，經過這次作業真正的了解到FC、Conv、sigmoid、Swish的forward&backward的運算是怎樣運行的，且也知道如何去建構一個完整的Deep Learning Architecture，不只是大概知道背後的理論是甚麼，而是可以用python去實現出來，雖然我的準確率非常的低...，我有想過可能是 Loss Function 的設定不好、Data preprocessing做得不好、Learning rate設的不夠小等等許多原因，如果可以的話可以根據我哪個環節問題給出回饋嗎？成績不在意模型的表現，但我很想知道哪裡出現了問題導致準確率超低。

## REFERENCES

- [1] <https://github.com/toxtli/lenet-5-mnist-from-scratch-numpy>  
[2] <https://github.com/vcoz17/Back-Propagation-in-CNN>.