**Deep-Learning-for-Autonomous-Driving**

**Lab1 : back propagation**

ID : 310605007

Member : 鄭晴立

Department : 機器人學程

目錄

[表目錄 2](#_Toc115625995)

[圖目錄 2](#_Toc115625996)

[1. Introduction 3](#_Toc115625997)

[2. Experiment setups 4](#_Toc115625998)

[2.1. Activation Functions 4](#_Toc115625999)

[2.2. Neural network 5](#_Toc115626000)

[2.3. Backpropagation and Optimizer 6](#_Toc115626001)

[2.4. Data augmentation 7](#_Toc115626002)

[3. Results of your testing 8](#_Toc115626003)

[4. Discussion 9](#_Toc115626004)

[4.1. Data augmentation 9](#_Toc115626005)

[4.2. 不同learning rate 9](#_Toc115626006)

[4.3. 將模型加深加大 10](#_Toc115626007)

[4.4. 不同的Batch size 11](#_Toc115626008)

[4.5. Sigmoid Function 12](#_Toc115626009)

# 表目錄

[表 一 Neural Network相關資料 3](#_Toc115625973)

[表 二 最終模型參數 8](#_Toc115625974)

# 圖目錄

[圖 一 Sigmoid functions code 4](#_Toc115625975)

[圖 二 Softmax functions code 4](#_Toc115625976)

[圖 三 Relu functions code 5](#_Toc115625977)

[圖 四 利用numpy創建三層Network 5](#_Toc115625978)

[圖 五 所有layer 的Backpropagation 6](#_Toc115625979)

[圖 六 利用Optimizer 更新參數 6](#_Toc115625980)

[圖 七 Data augmentation code 7](#_Toc115625981)

[圖 八 Data augmentation 7](#_Toc115625982)

[圖 九 Learning rate = 0.005 9](#_Toc115625983)

[圖 十 Learning rate = 0.01 10](#_Toc115625984)

[圖 十一 Learning rate = 0.1 10](#_Toc115625985)

[圖 十二 加大網路 10](#_Toc115625986)

[圖 十三加深網路 11](#_Toc115625987)

[圖 十四 batch size = 1024 11](#_Toc115625988)

[圖 十五batch size = 512 11](#_Toc115625989)

[圖 十六 全部改為sigmol learning rate 為0.0001 12](#_Toc115625990)

[圖 十七 全部改為sigmol learning rate 為0.01 12](#_Toc115625991)

[圖 十八2兩層改為sigmol learning rate 為0.01 12](#_Toc115625992)

# Introduction

本次實驗要分辨47類28\*28的圖片，並用numpy去實作網路，程式部分大多按照助教所給提示及檔案去完成，盡量不去做大幅度的修改。

表 一 Neural Network相關資料

|  |  |
| --- | --- |
| Parameter | Neural Network |
| Number of input layer | 784 |
| Number of hidden layer | 2~5 |
| Number of output layer | 47 |
| Activation function | Relu、Sigmoid、Softmax |
| Optimizer | Adam、SDG |
| Learning rate | 0.01~0.0001 |
| Epoch | 50~1000 |
| Loss function | Mean Squared Error |
| Batch size | 256~1024 |

# Experiment setups

## Activation Functions

本實驗最後一層採用softmax，其餘則嘗試relu及、sigmoid，由於本次為47分類的圖像辨識問題，relu的結果應該會較佳，最後與預期符合，以下為Activation Functions的說明與實作。

一張含有 文字 的圖片

自動產生的描述

圖 一 Sigmoid functions code

一張含有 文字 的圖片

自動產生的描述

圖 二 Softmax functions code

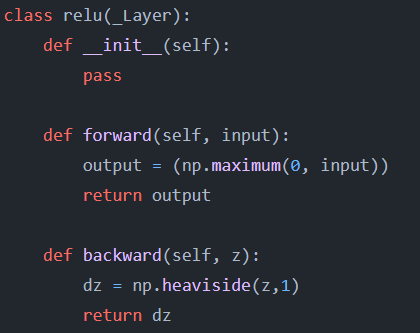


圖 三 Relu functions code

## Neural network

本實驗使用三到五層網路，其模型示意圖，經過實驗發現，加深網路及加大網路，皆無法使結果更好，其實程式說明如圖 四。

一張含有 文字 的圖片

自動產生的描述

圖 四 利用numpy創建三層Network

## Backpropagation and Optimizer

有別於助教給的code，我把Optimizer 放在Backpropagation 後面，實作上相對容易，backward之後直接更新參數，其程式如圖 五、圖 六。

一張含有 文字 的圖片

自動產生的描述

圖 五 所有layer 的Backpropagation

一張含有 文字 的圖片

自動產生的描述

圖 六 利用Optimizer 更新參數

## Data augmentation

我嘗試使用Data augmentation 去改善overfitting，我將圖片往八個角落移動1個pixel，使我的資料有九種變化，最終使精準度有效提升。

一張含有 文字 的圖片

自動產生的描述

圖 七 Data augmentation code

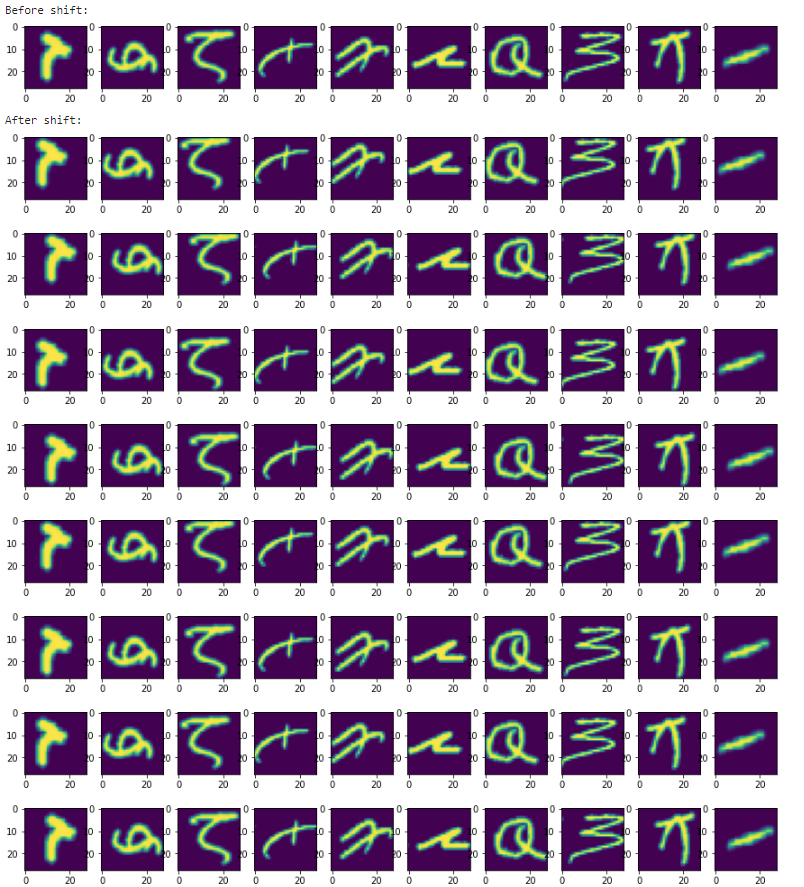


圖 八 Data augmentation

# Results of your testing

經過反覆測試，利用此模型，最高可達86.679%左右的精確度。

表 二 最終模型參數

|  |  |
| --- | --- |
| Parameter | Neural Network |
| Number of input layer | 784 |
| Number of hidden layer | 3 |
| Number of output layer | 47 |
| Activation function | Relu、Softmax |
| Optimizer | Adam |
| Learning rate | 0.0001 |
| Epoch | 500 |
| Loss function | Mean Squared Error |
| Batch size | 256 |

一張含有 文字 的圖片

自動產生的描述

圖 九 最高Accuracy

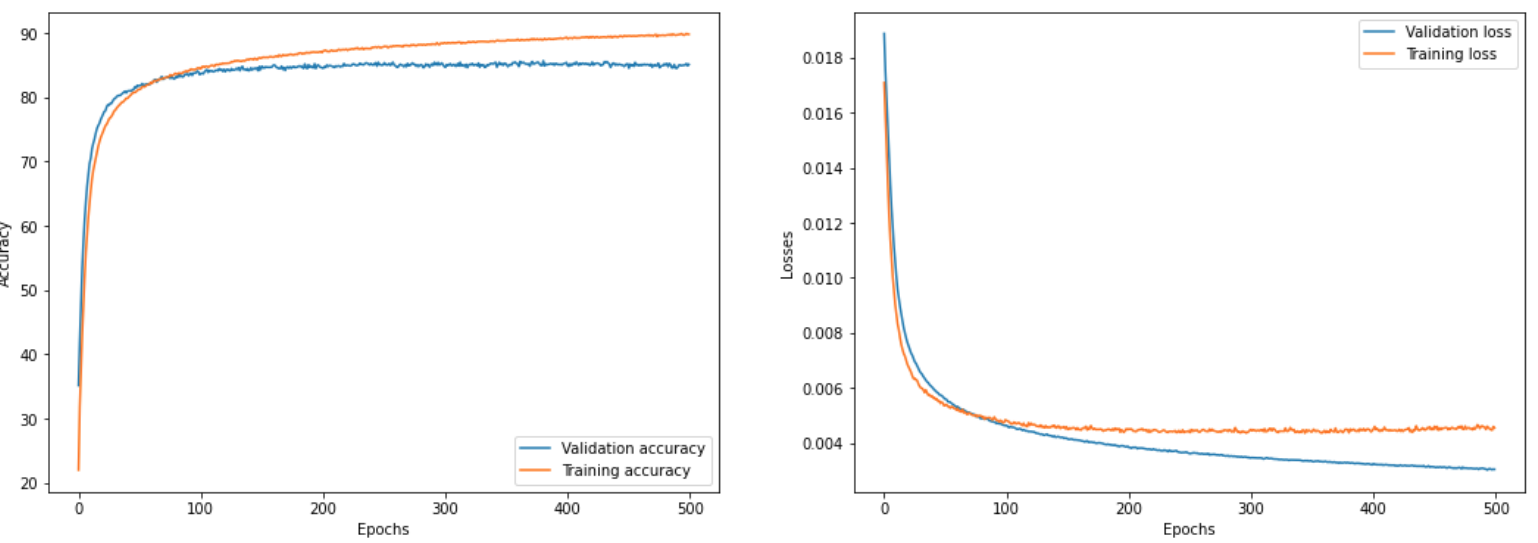


圖 十 Loss及Accuracy

# Discussion

因為時間關係，大多沒有完成訓練，僅訓練到大致看得出結果為止。

## Data augmentation

大幅改善overfitting的問題。

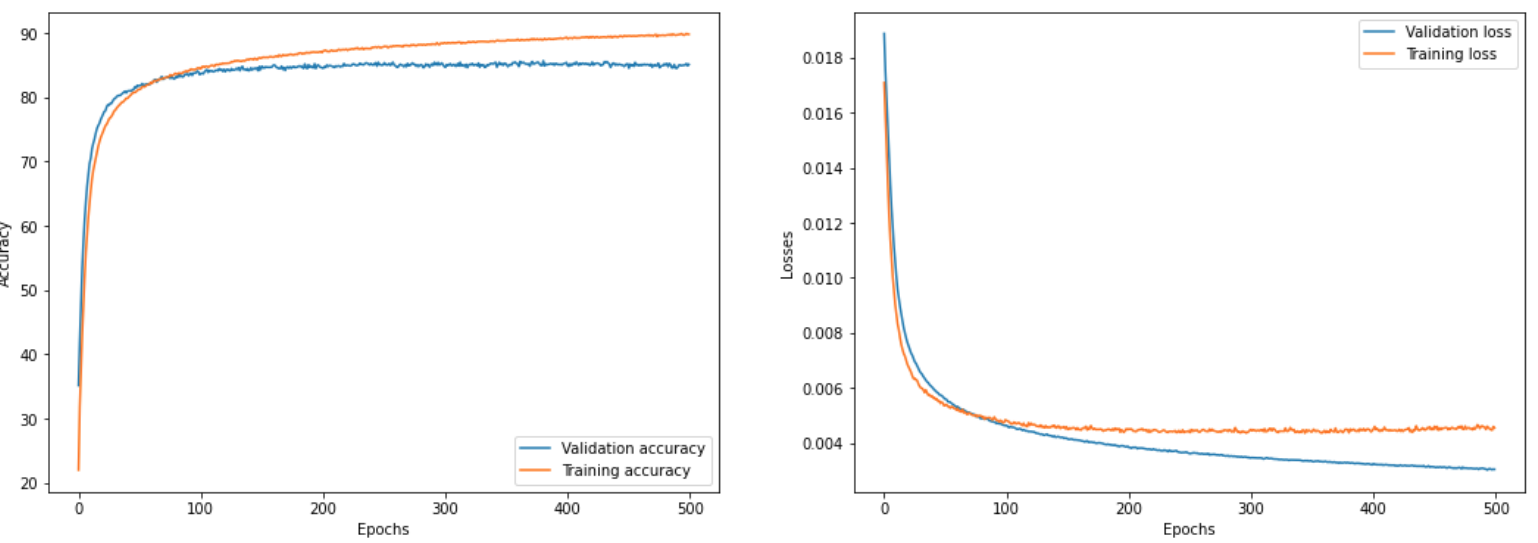


圖 十一利用Data augmentation去改善模型

## 不同learning rate

我嘗試了不同的learning rate，在learning rate>0.001之後，Accuracy幾乎都很小，很快就發生，也很容易趨向發生overfitting。

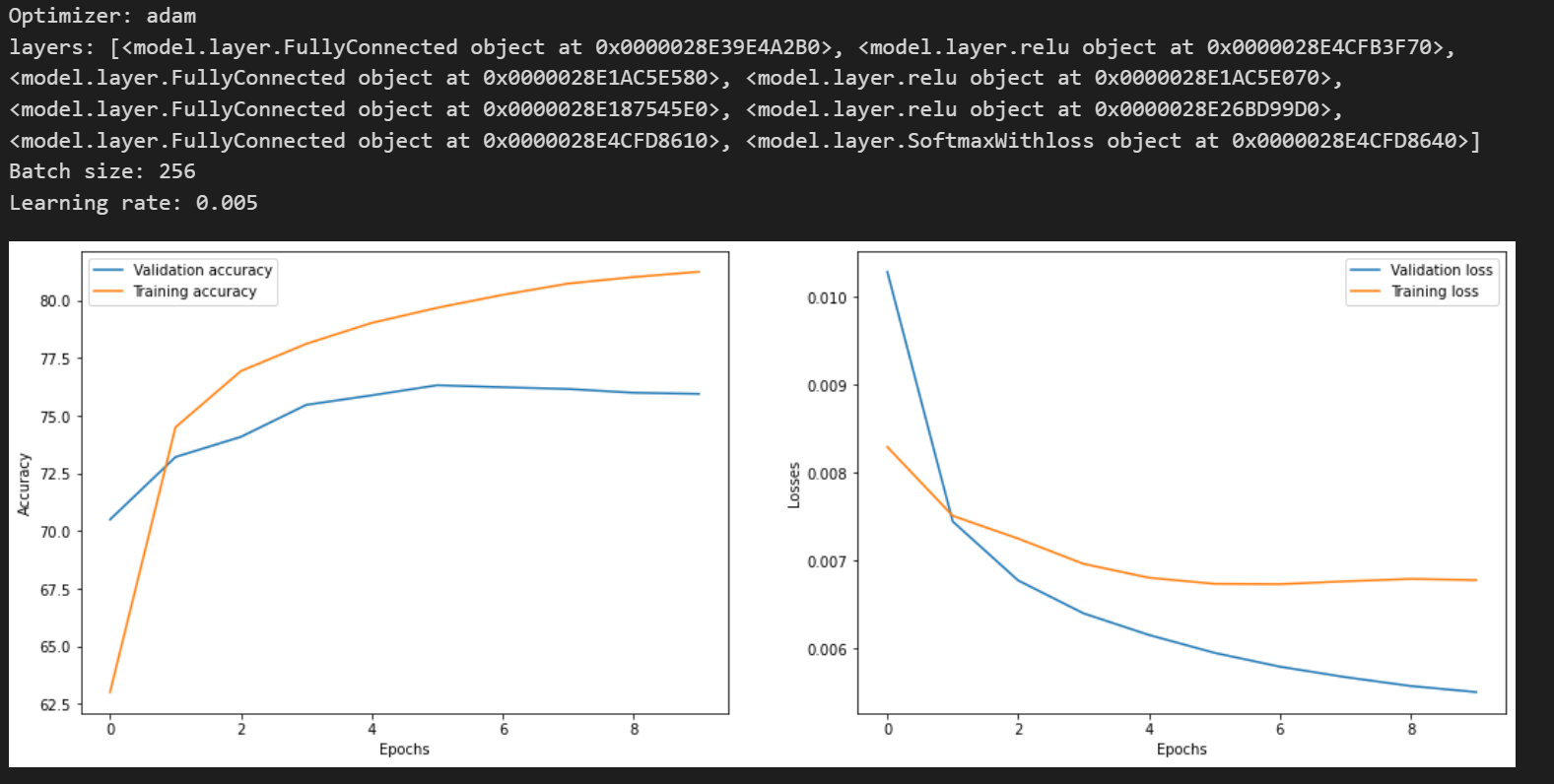


圖 十二 Learning rate = 0.005

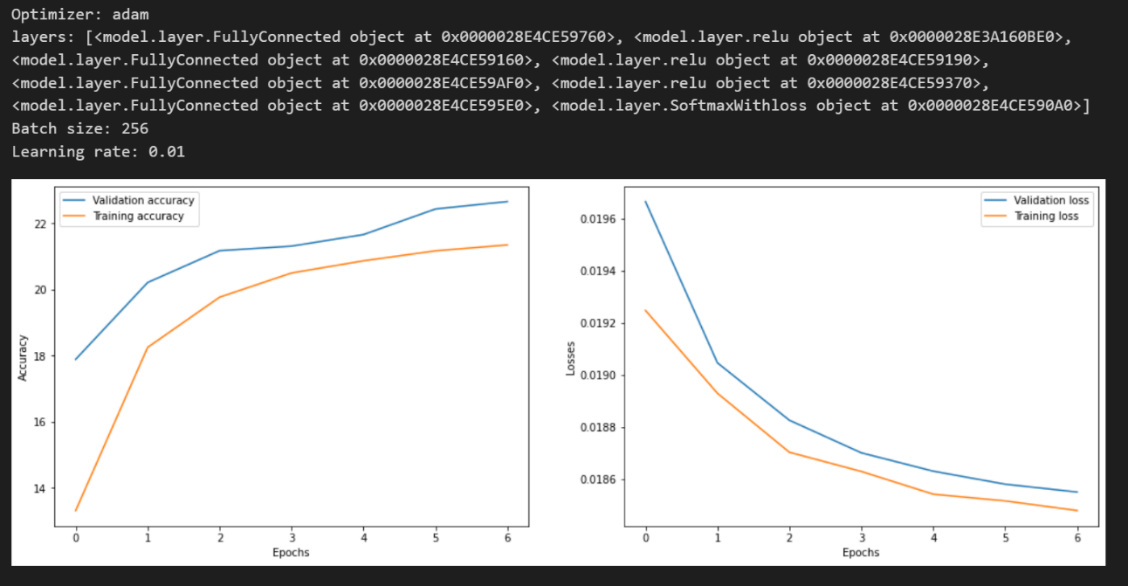


圖 十三 Learning rate = 0.01

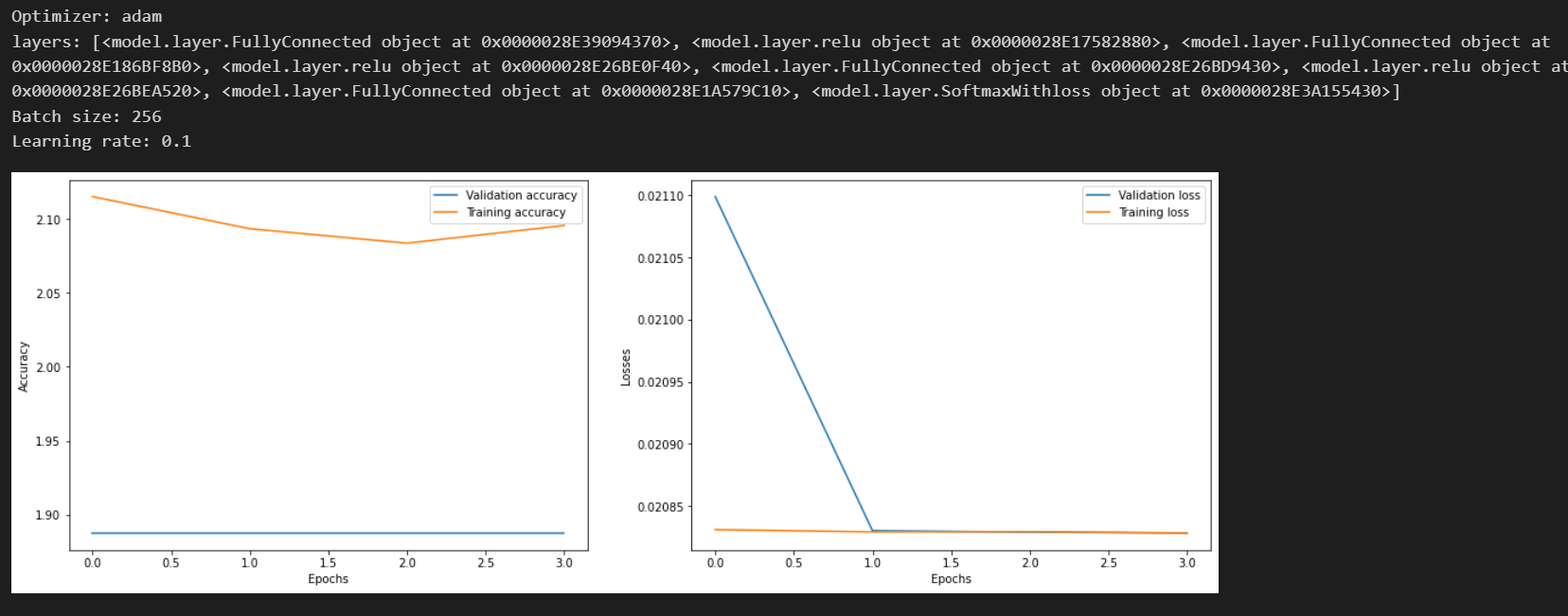


圖 十四 Learning rate = 0.1

## 將模型加深加大

嘗試將模型加大加深，也是非常容易overfitting，但結果相對好。

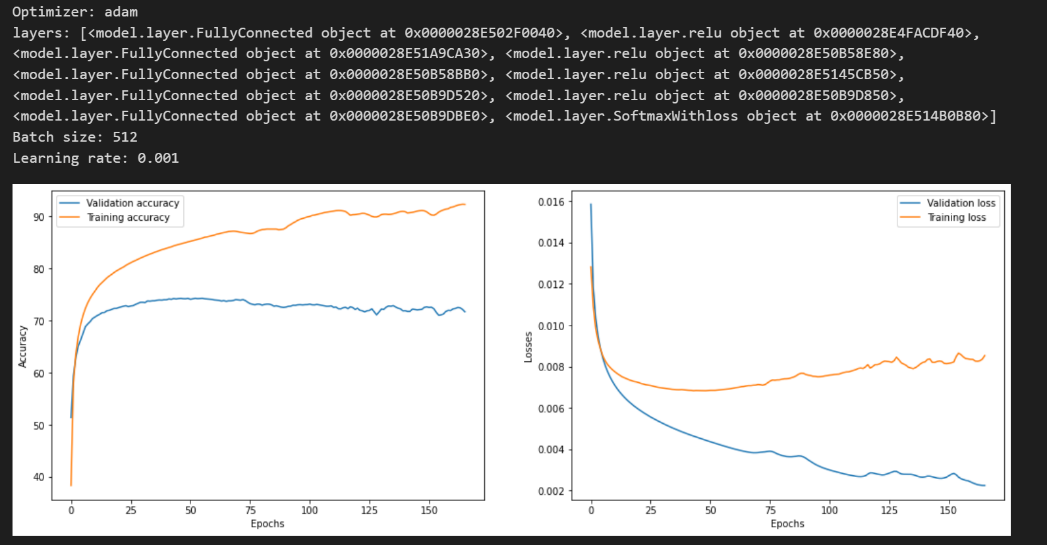


圖 十五 加大網路

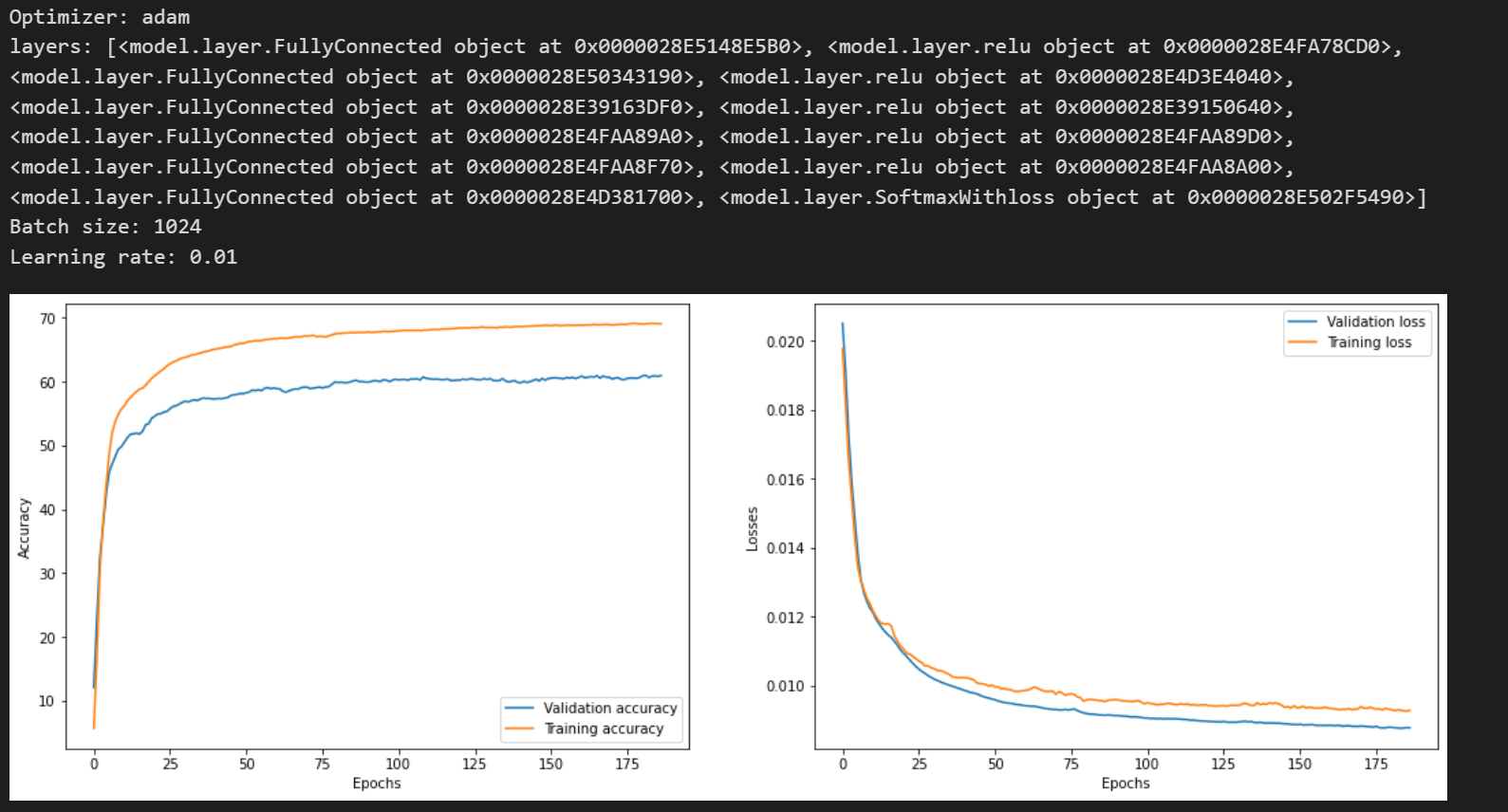


圖 十六加深網路

## 不同的Batch size

經過多次嘗試，不同batch size對結果影響不大，主要還是在模型架構的調整。

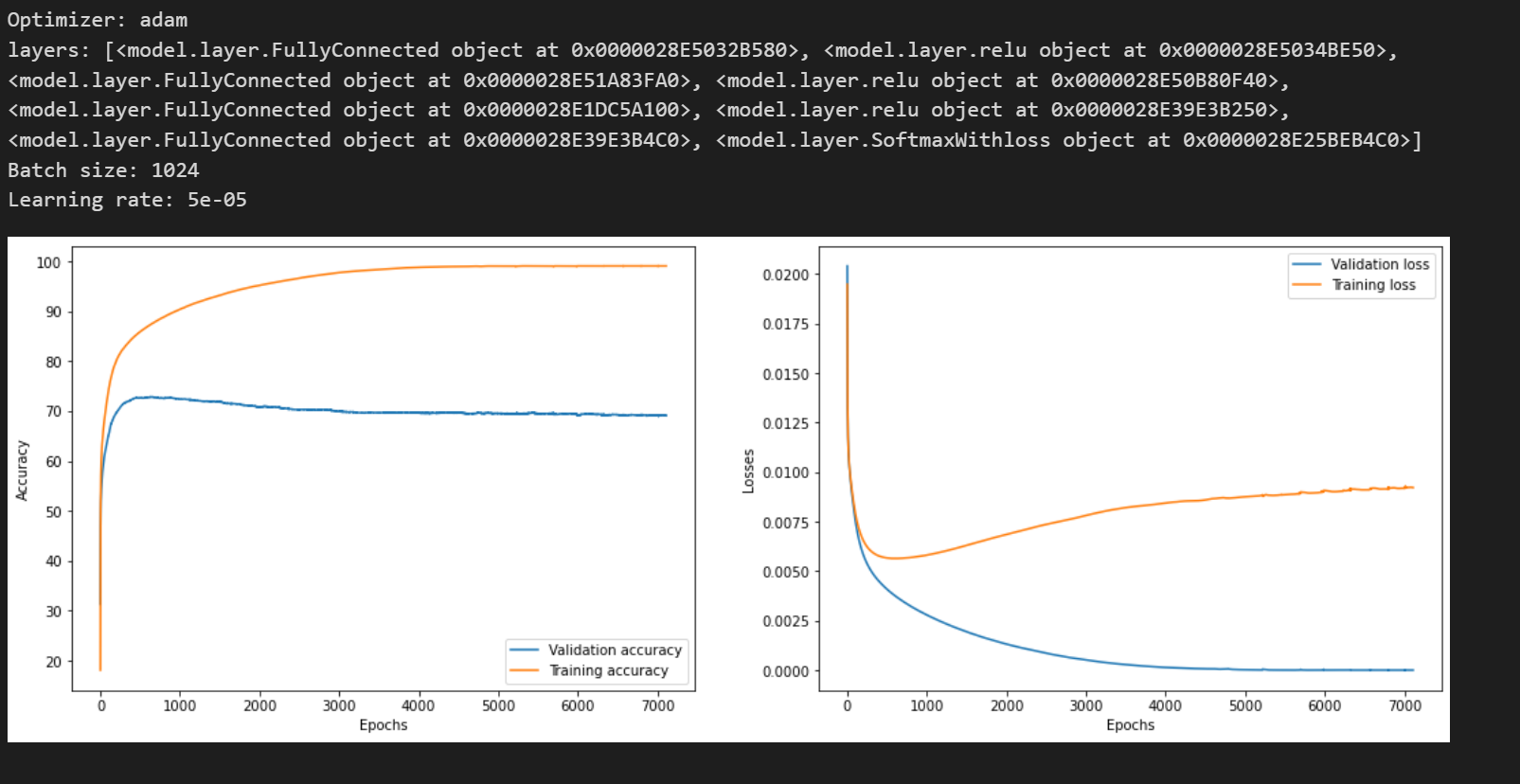


圖 十七 batch size = 1024

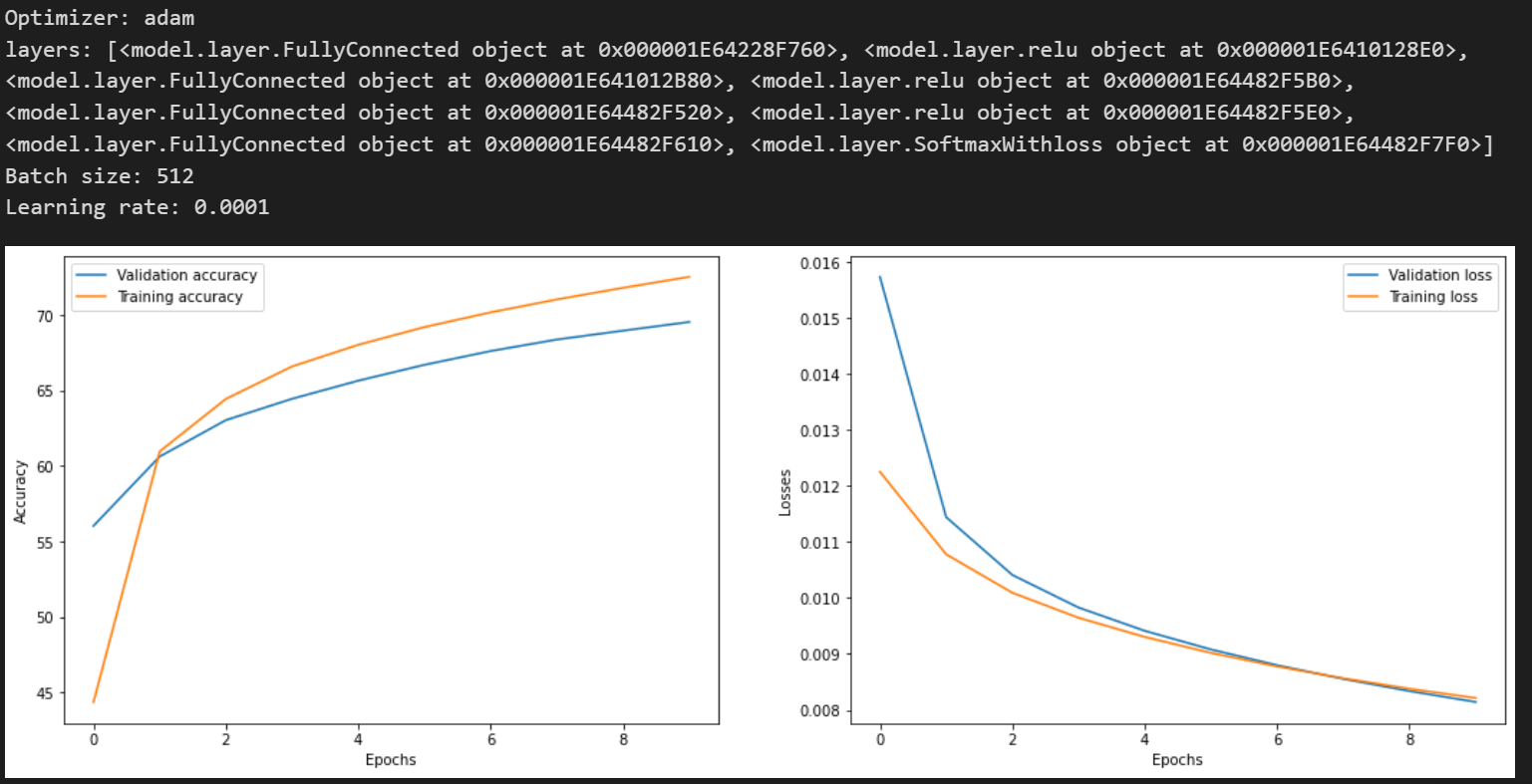


圖 十八batch size = 512

## Sigmoid Function

嘗試使用sigmiod 的結果，accuracy皆無法有效提升，因此，最後使用relu作為Activation Function。

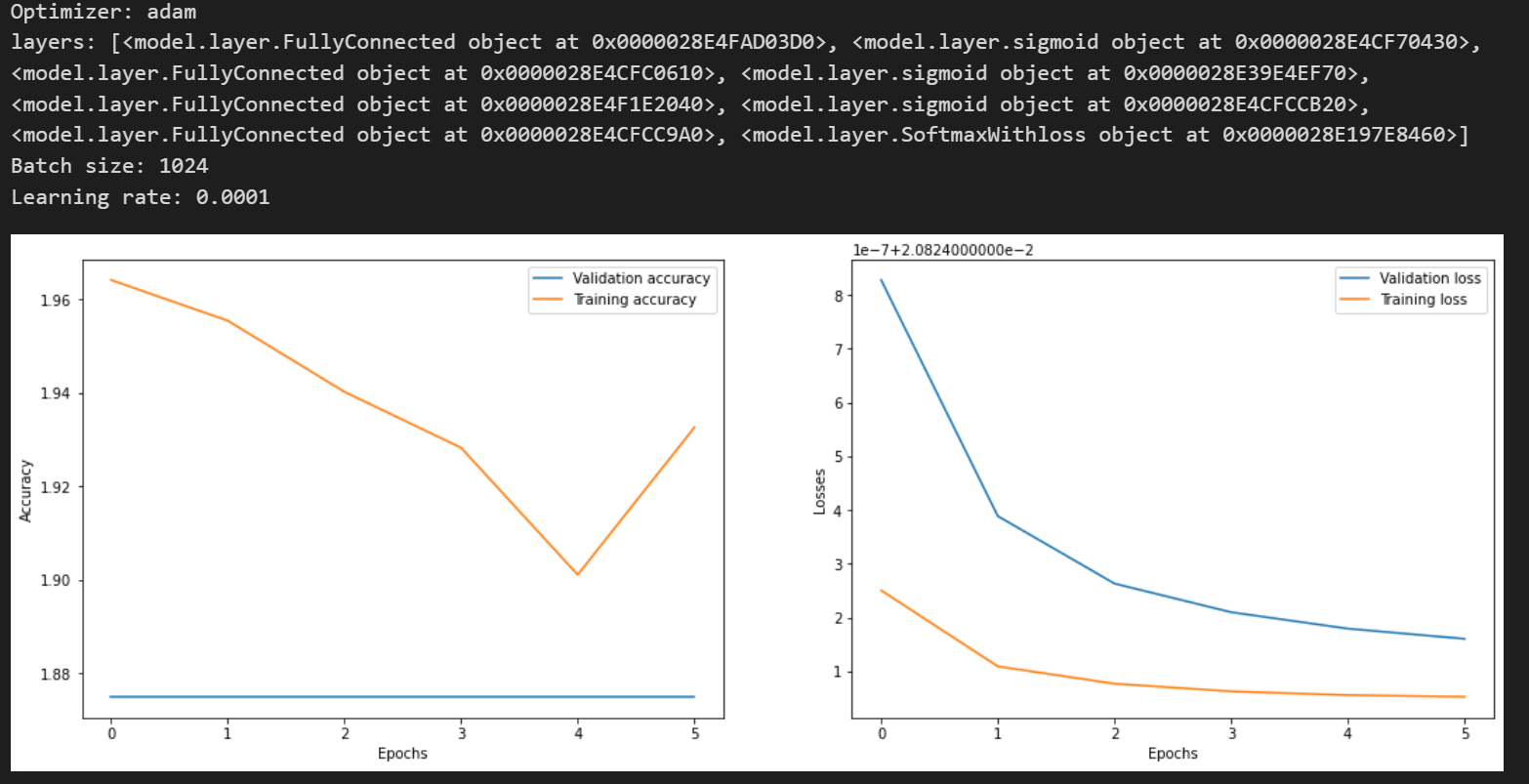


圖 十九 全部改為sigmol learning rate 為0.0001

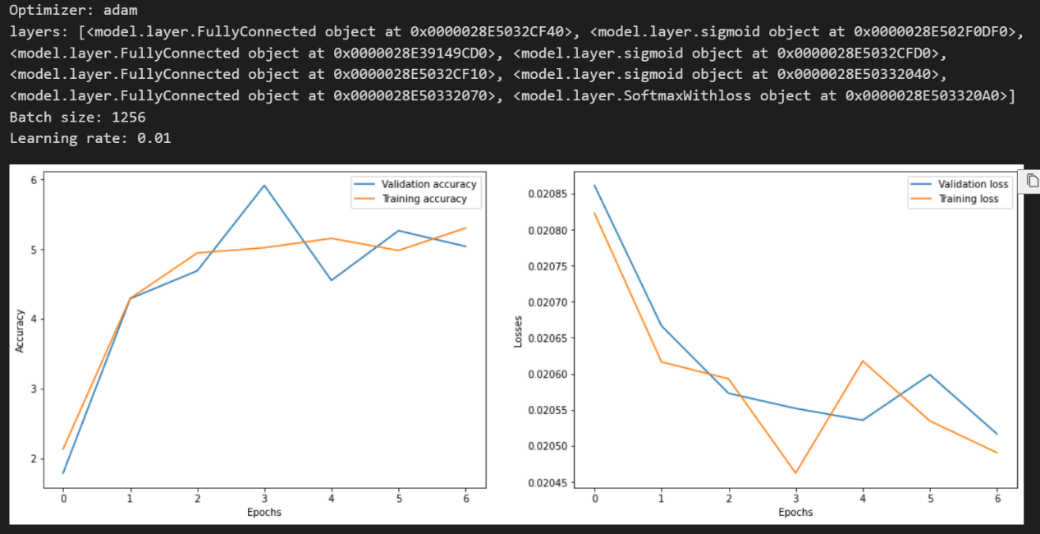


圖 二十 全部改為sigmol learning rate 為0.01

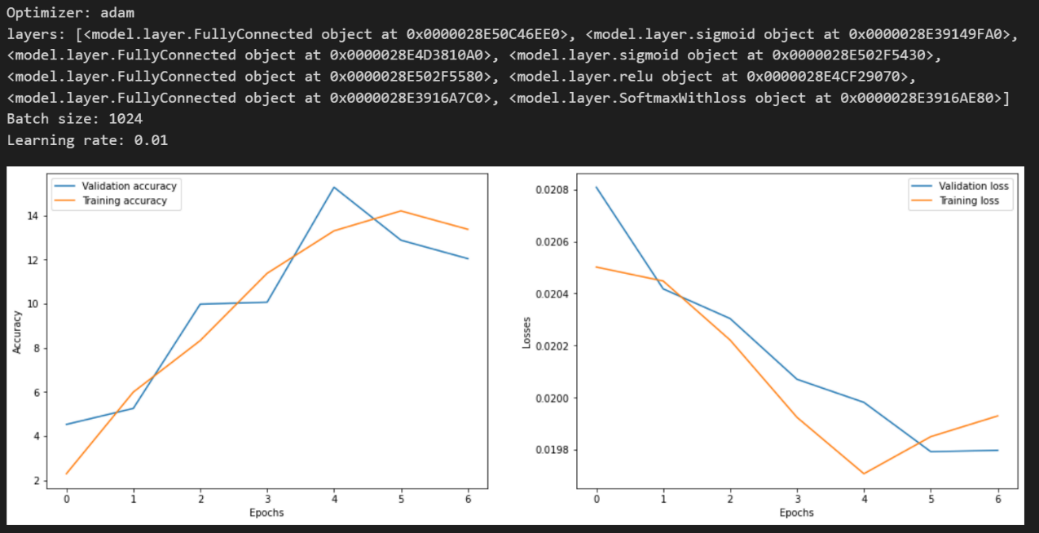


圖 二十一2兩層改為sigmol learning rate 為0.01