

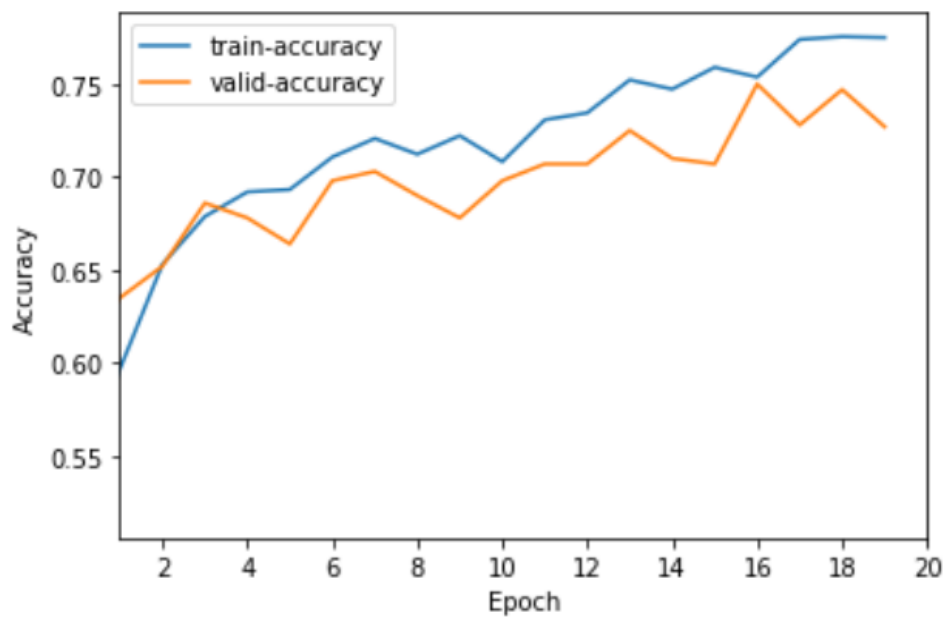
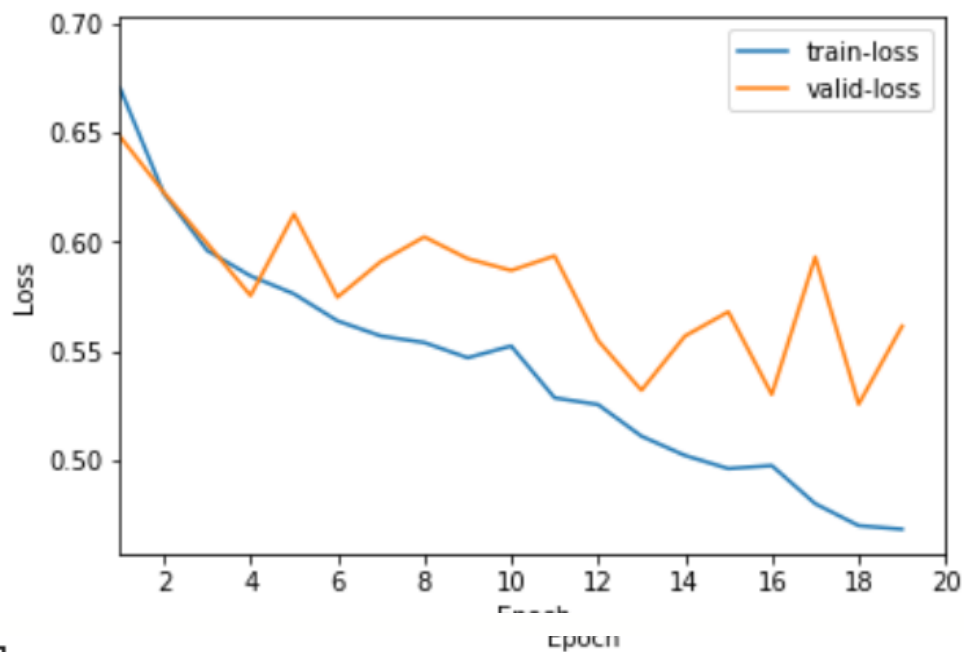
## 作業要求-1

(以下皆為在google colab上所訓練的結果)

1.

```
train_batch_size = 32  
epochs = 20  
learning_rate = 0.01
```

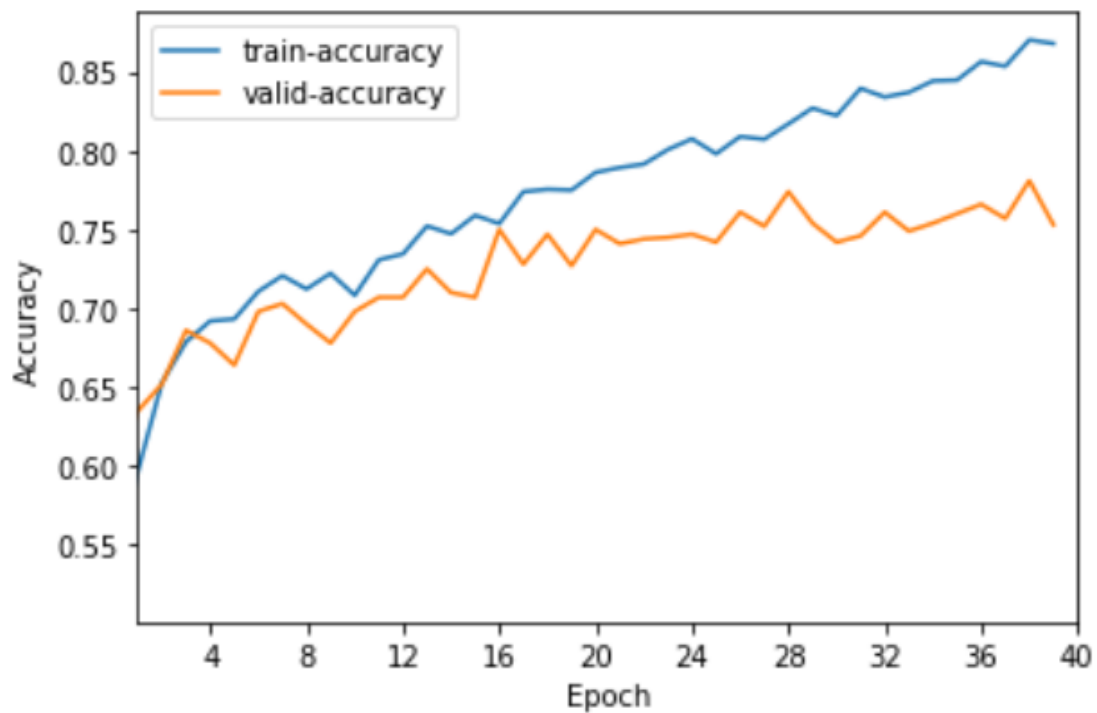
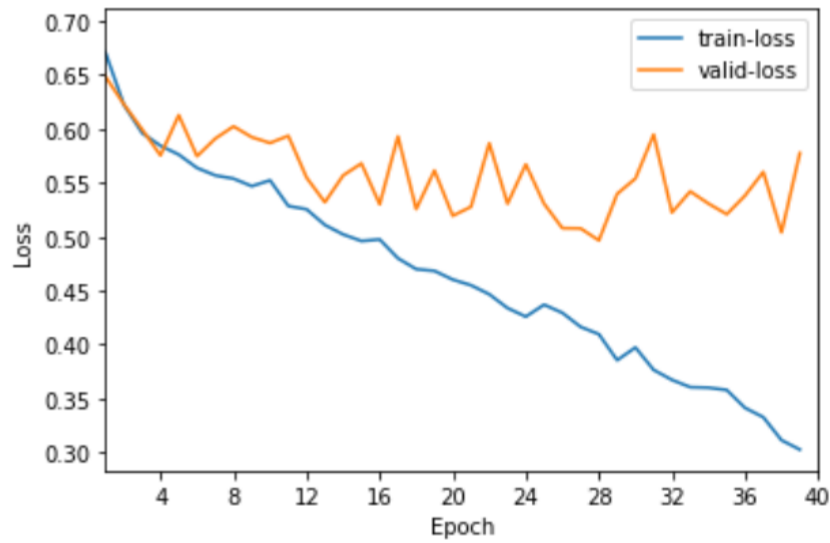
Finished Training



Testing: 100% ██████████ 500/500 [00:05<00:00, 85.13it/s]Test accuracy: 75.4000%

```
train_batch_size = 32
epochs = 40
learning_rate = 0.01
```

Finished Training

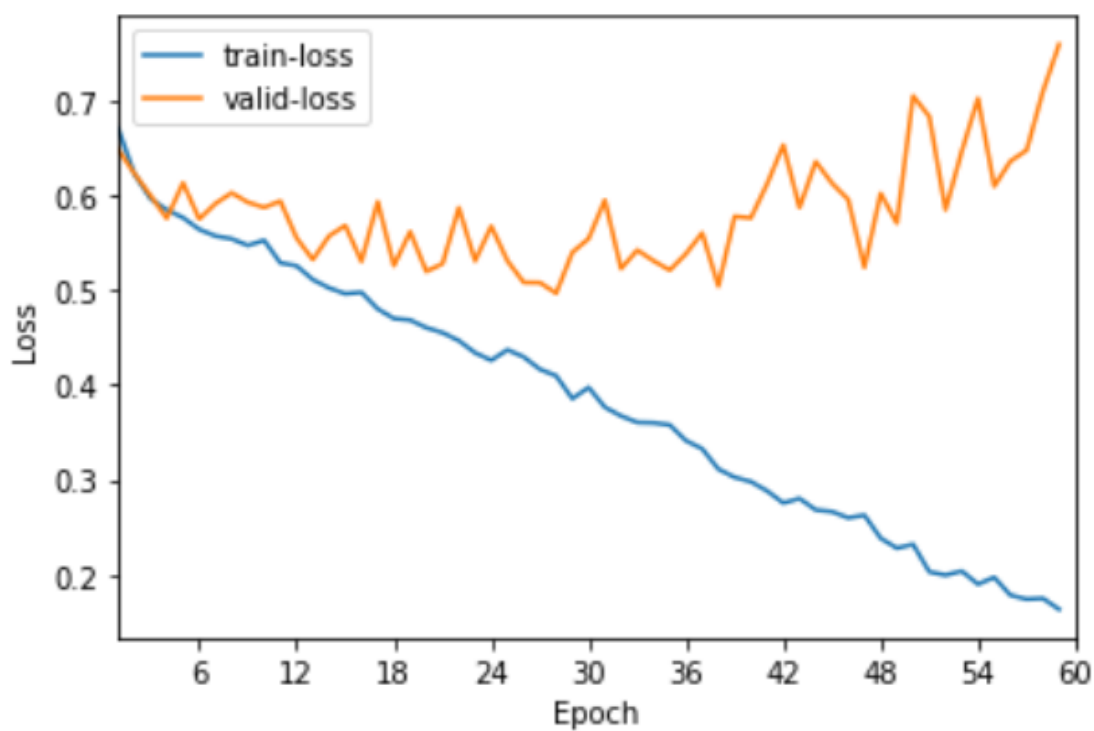


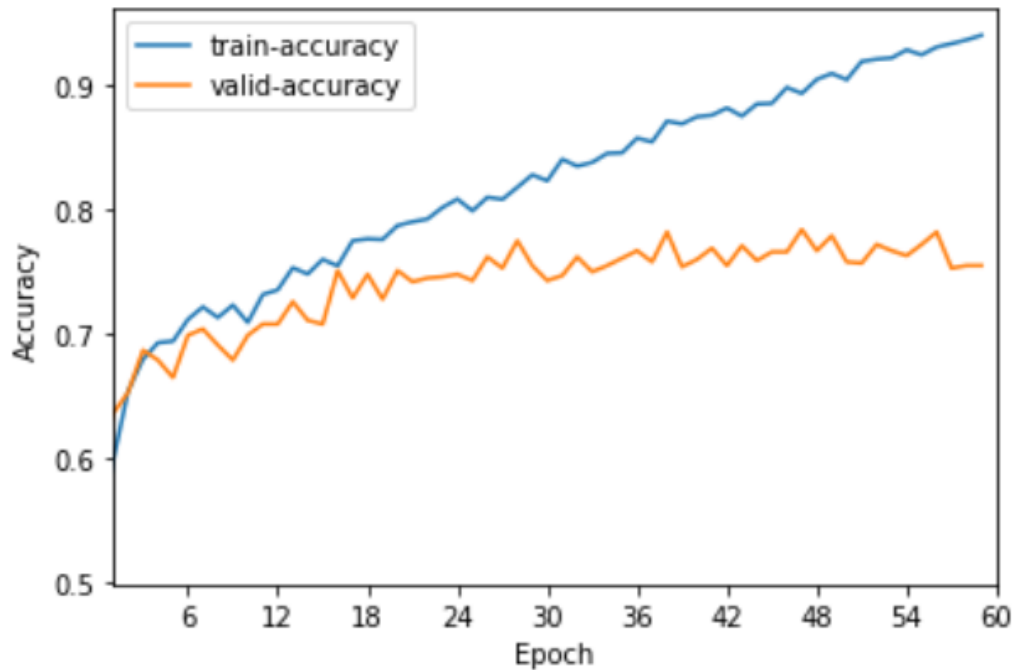
---

Testing: 100% ██████████ 500/500 [04:20<00:00, 1.92it/s]Test accuracy: 73.9000%

```
train_batch_size = 32  
epochs = 60  
learning_rate = 0.01
```

## Finished Training





Testing: 100% [████████████████████] 500/500 [00:06<00:00, 83.29it/s] Test accuracy: 73.9000%

隨著epoch上升, train\_loss最終的值也會隨之下降, epoch=60時, train\_loss值為三者最小, 而反之, valid\_loss則是隨之上升, epoch=60時, valid\_loss為三者之中最大。

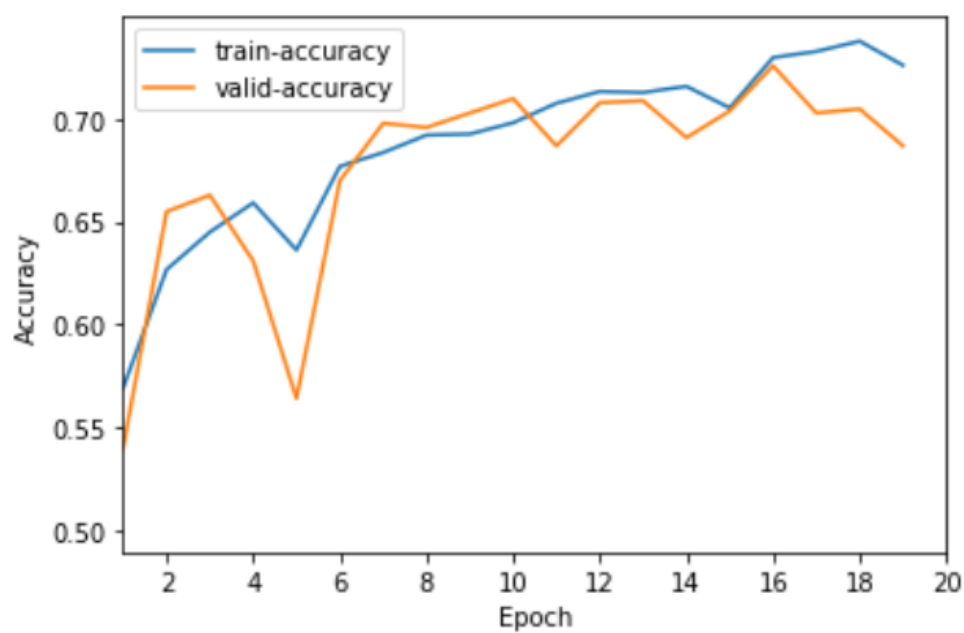
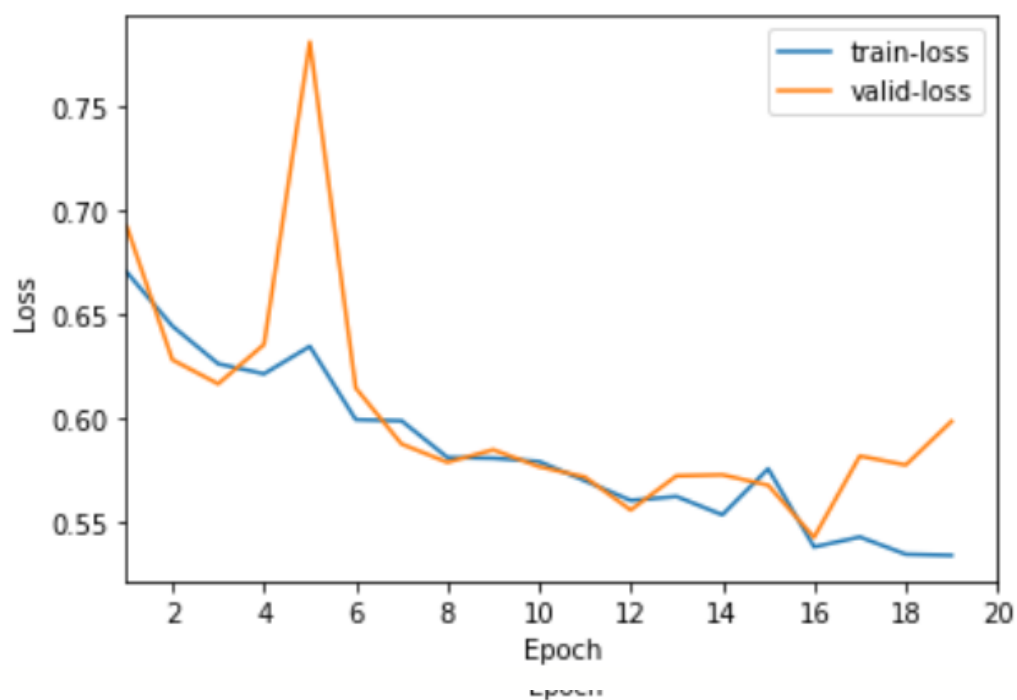
當epoch上升時, train\_accuracy值同時也隨之上升(準確率越高), 當epoch=60時, 可發現 train\_accuracy為最大(約0.95), 而當epoch=20時, train\_accuracy介於0.8~0.75之間。而valid\_accuracy, 則不隨epoch而有所改變, 無論epoch=60、40還是20, valid\_accuracy都約介於0.7~0.75之間。

也可發現當epoch上升時, 兩條線彼此是漸漸遠離的, 在epoch=60時最為明顯。

2.

```
train_batch_size = 8
epochs = 20
learning_rate = 0.01
```

## Finished Training

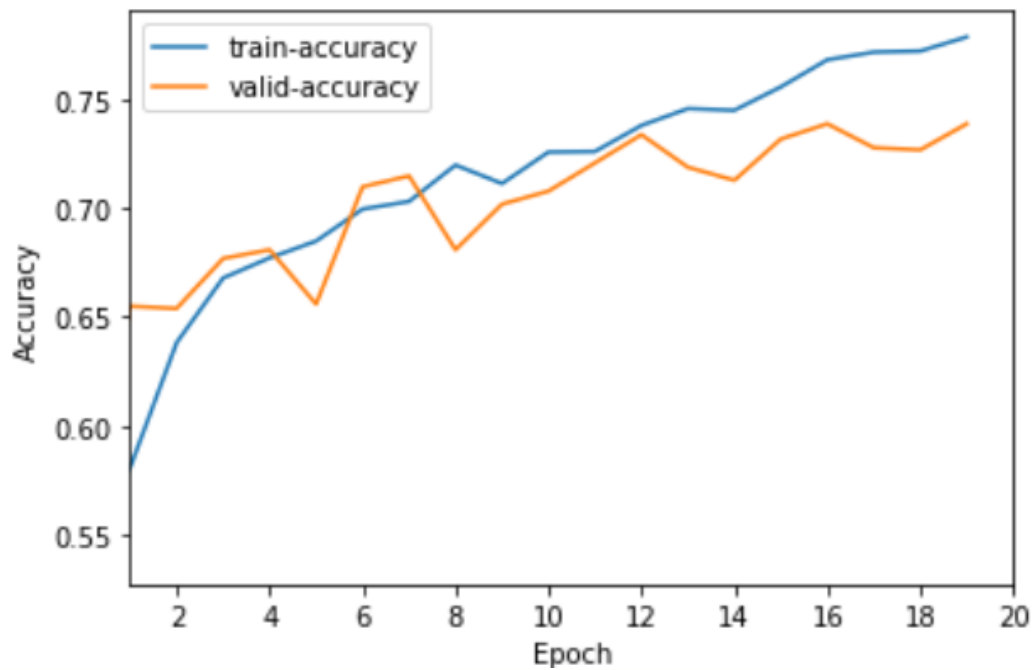
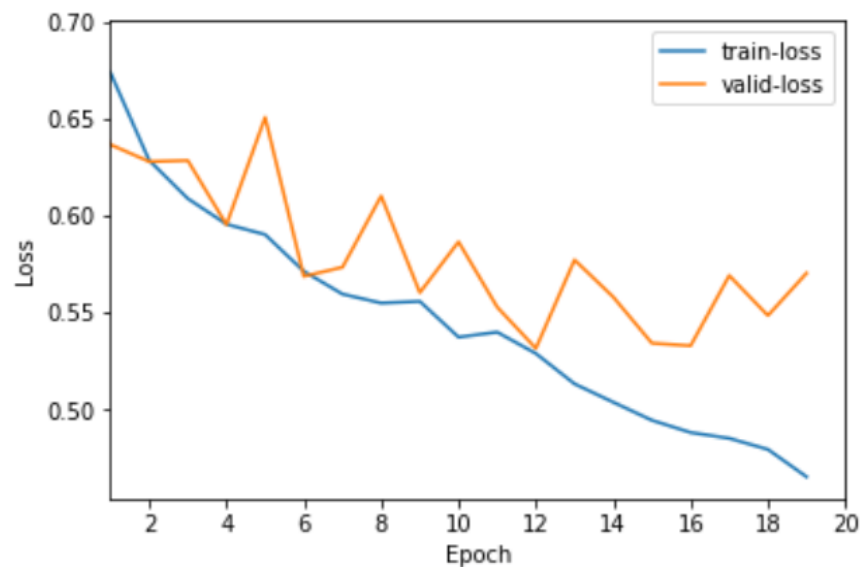


Testing: 100% |████████████████████| 500/500 [00:05<00:00, 84.75it/s] Test accuracy: 68.1000%

```
train_batch_size = 16
epochs = 20
learning_rate = 0.01
```



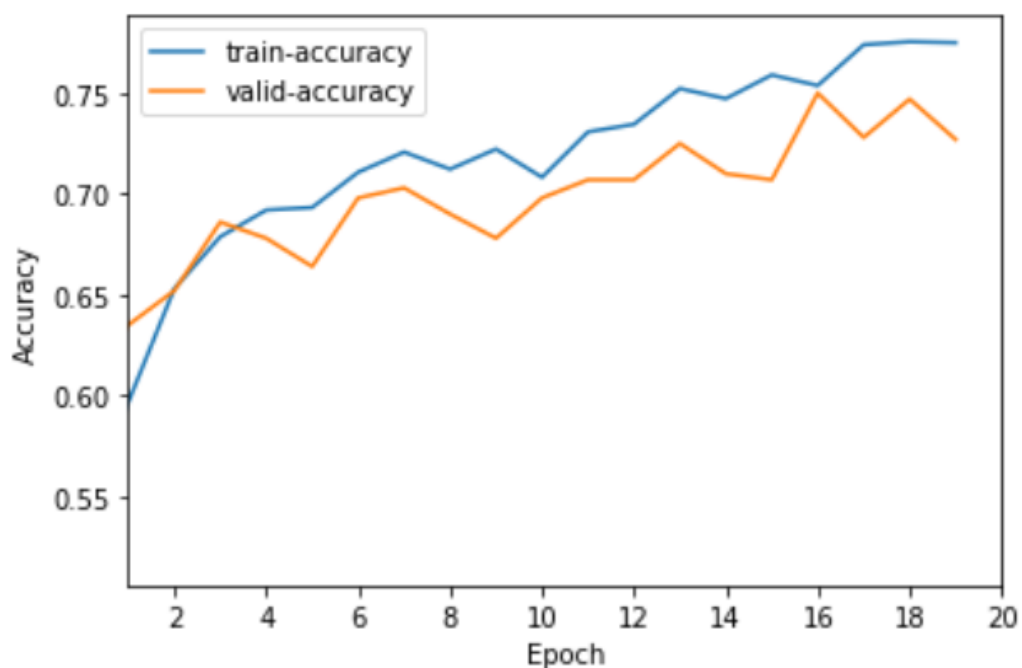
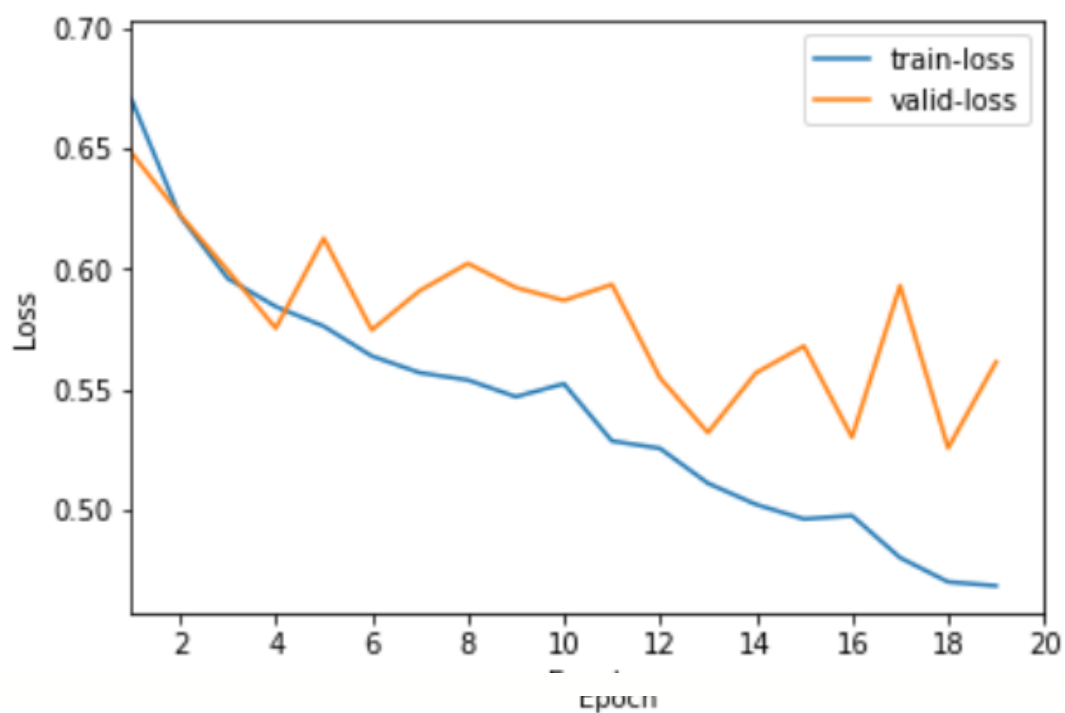
## Finished Training



Testing: 100% ██████████ 500/500 [00:05<00:00, 83.90it/s] Test accuracy: 71.2000%

```
train_batch_size = 32
epochs = 20
learning_rate = 0.01
```

## Finished Training



Testing: 100% ██████████ 500/500 [00:05<00:00, 85.17it/s] Test accuracy: 75.4000%

batch\_size對於train\_loss及valid\_loss, 只有在batch\_size從8到16這段, 兩者最終的值有些許的改變, 而從16到32兩者最終的值並未有太大的差異, 但從圖中發現batch\_size上升時, 兩條曲線契合程度會下降, 像是batch\_size=8時, 兩者在epoch=16前近乎重疊, 而在batch\_size上升後, 可發現兩者曲線彼此相距漸遠。

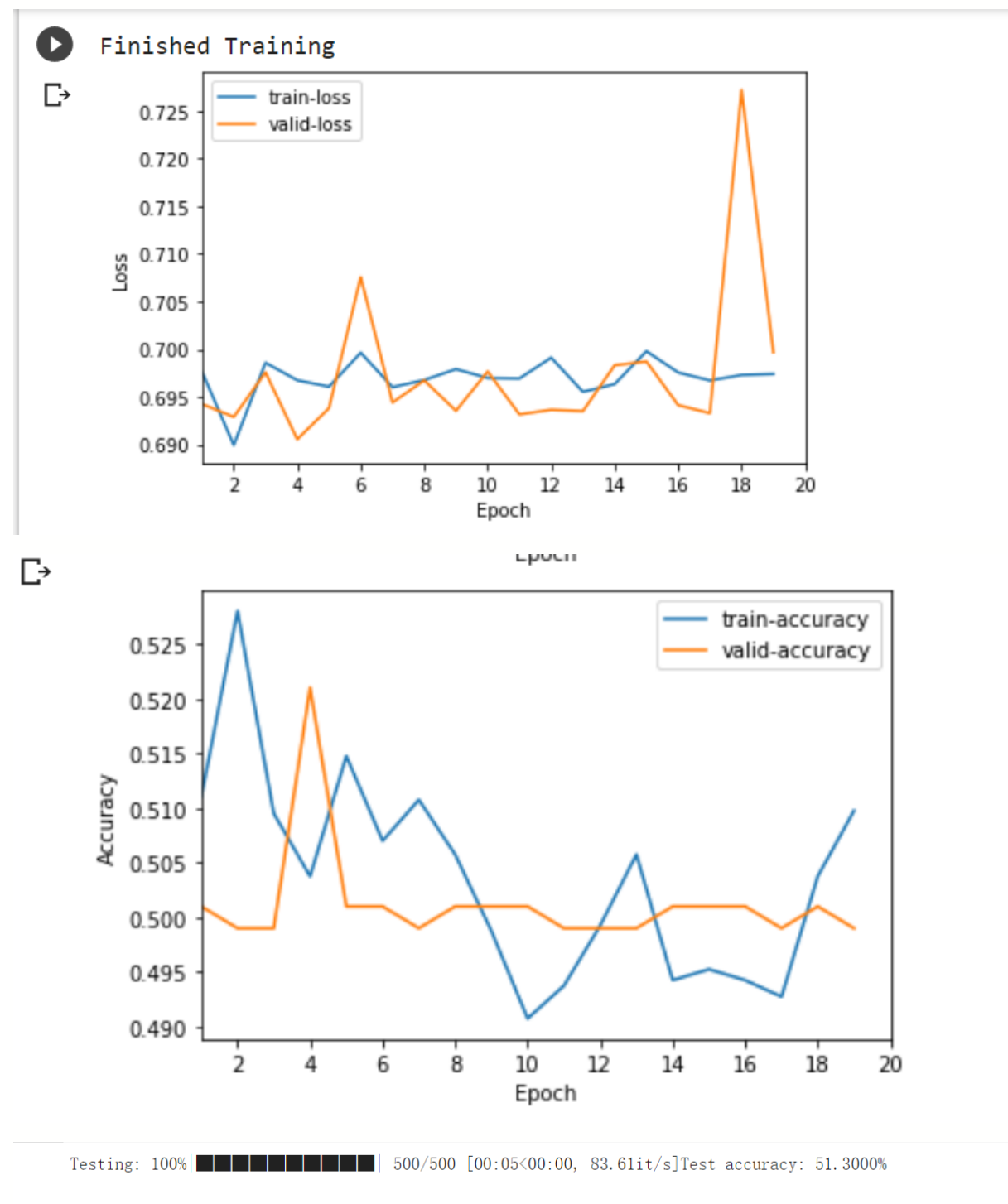
batch\_size對於train\_accuracy及valid\_accuracy, 跟loss的呈現差不多的規律, 在batch\_size從16到32時, 最終值並未有顯著的改變, 在batch\_size從8到16這段才有些許的提升, 而batch\_size同樣影響兩者accuracy的契合度, batch\_size上升, 兩者契合度越低。

3.

```
train_batch_size = 32
```

```
epochs = 20
```

```
learning_rate = 0.1
```



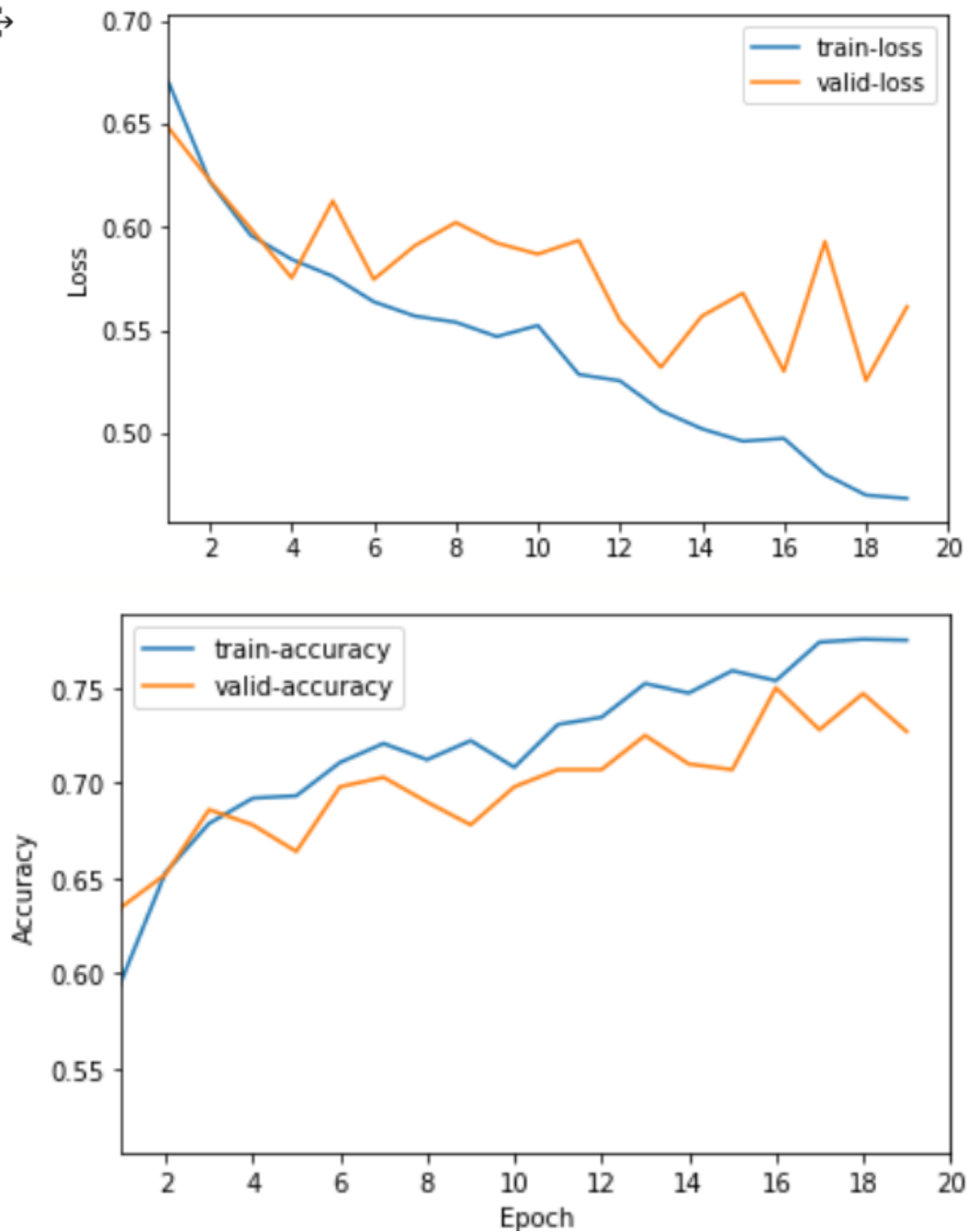
```
train_batch_size = 32
```



```
epochs = 20
learning_rate = 0.01
```



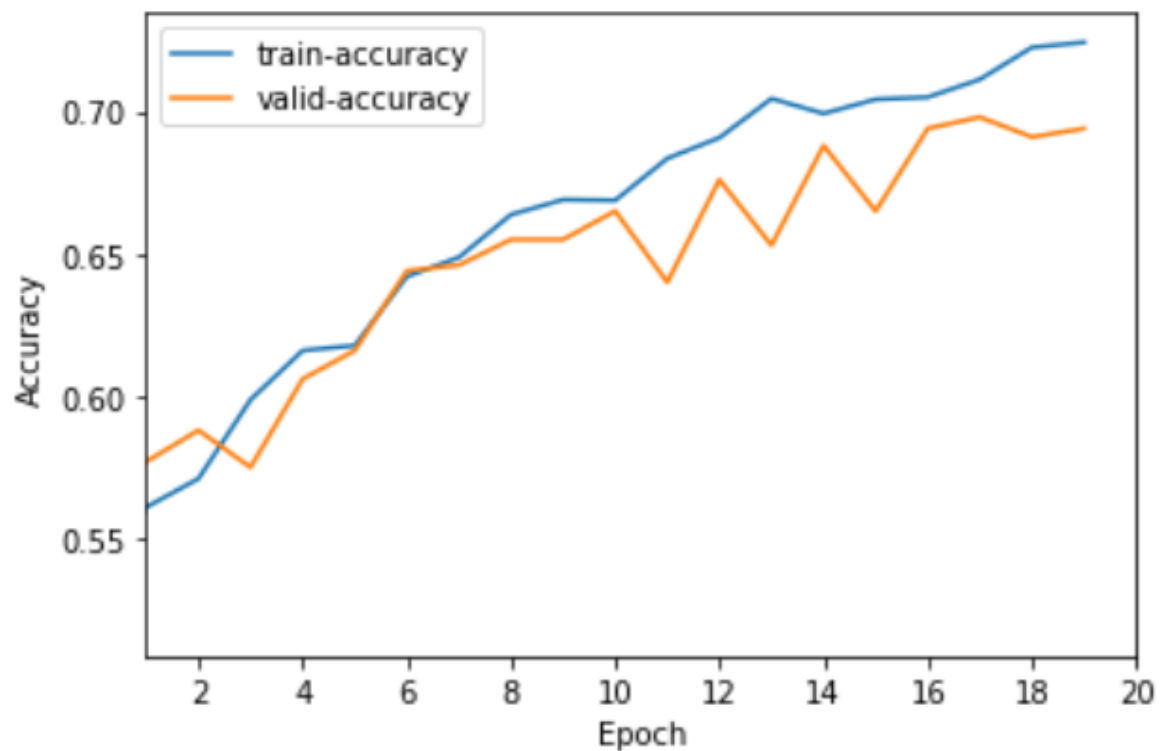
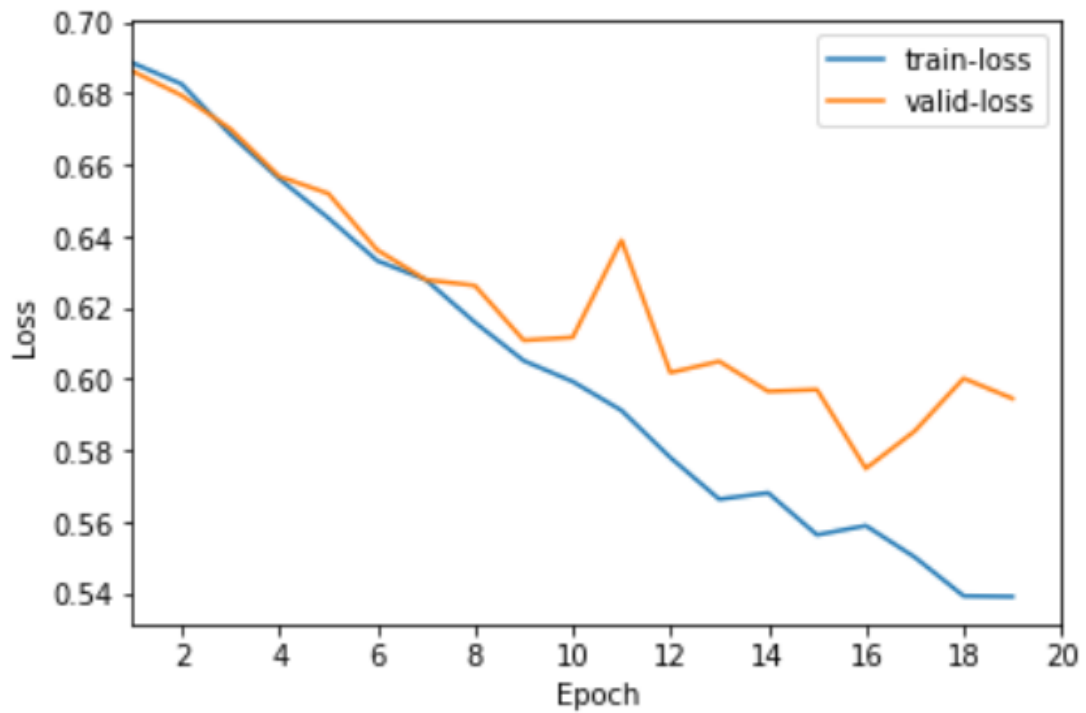
## Finished Training



Testing: 100% [████████████████████] 500/500 [00:05<00:00, 86.27it/s] Test accuracy: 75.4000%

```
train_batch_size = 32
epochs = 20
learning_rate = 0.001
```

## Finished Training



Testing: 100% |████████████████████| 500/500 [00:06<00:00, 80.69it/s] Test accuracy: 67.2000%

learning rate過高(0.1)時, train\_loss及valid\_loss在圖中呈現凌亂的走向, 代表訓練存在問題, 而當learning rate下降至0.01或0.001時, 可發現train\_loss及valid\_loss有下降的規律, 而0.001時, 下降的程度又更加平滑且規律。

當learning rate過高, 可明顯看出train\_accuracy及valid\_accuracy其走向並未有規律的在圖中移動, 代表訓練有誤, 而當learning rate下降至0.01時, 可發現兩者的accuracy呈現規律的走向, 但發現learning rate下降至0.001時, train\_accuracy的值比learning rate為0.01時較低, 代表在0.001時可能存在overfitting的疑慮。

## 作業要求-2

(以下皆為在google colab上所訓練的結果)

model修改的部分

將model第一層self.cnn1的out\_channel修改為32, 第二層的self.cnn2的in\_channel也改成32且kernel\_size改成2。

第三層self.cnn3的out\_channel改成32、kernel\_size改成4,

因為多加了第四層, 所以將前面三層的節點稍微減少, 改變out\_channel跟in\_channel的數目, 避免過度訓練, 並且在前三層的kernel\_size做些更動, 讓訓練較有彈性些。

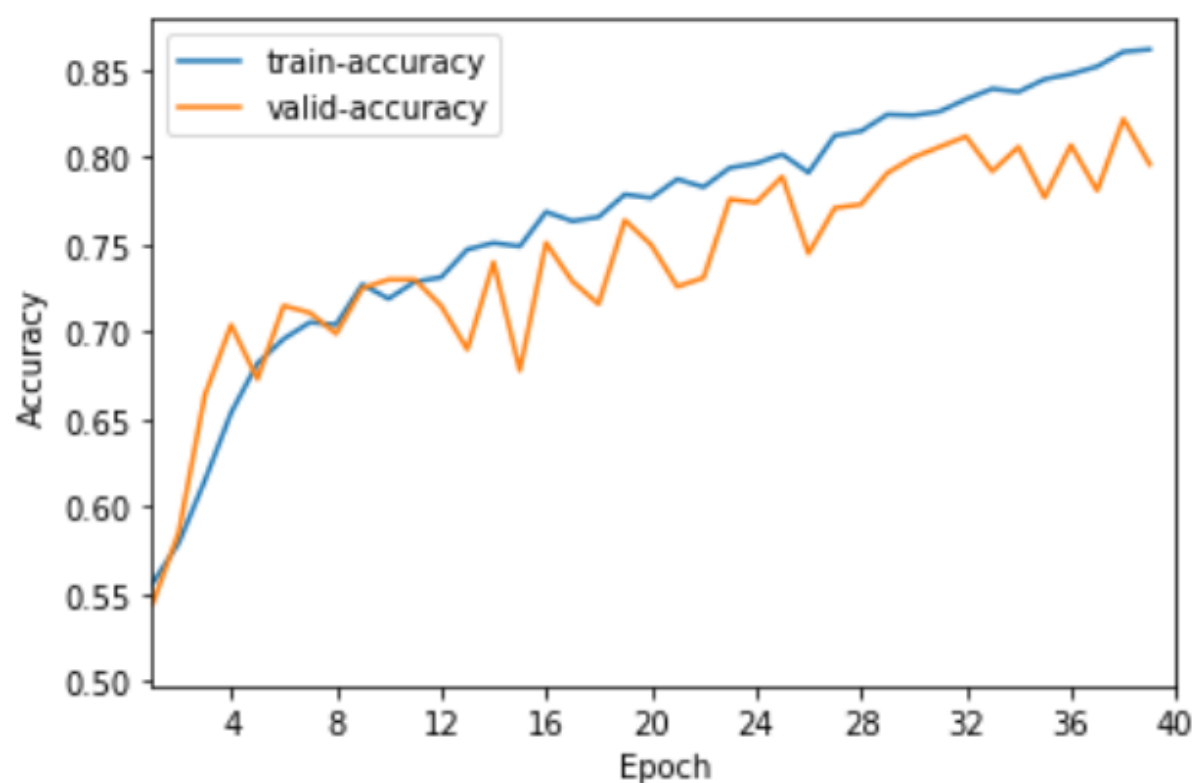
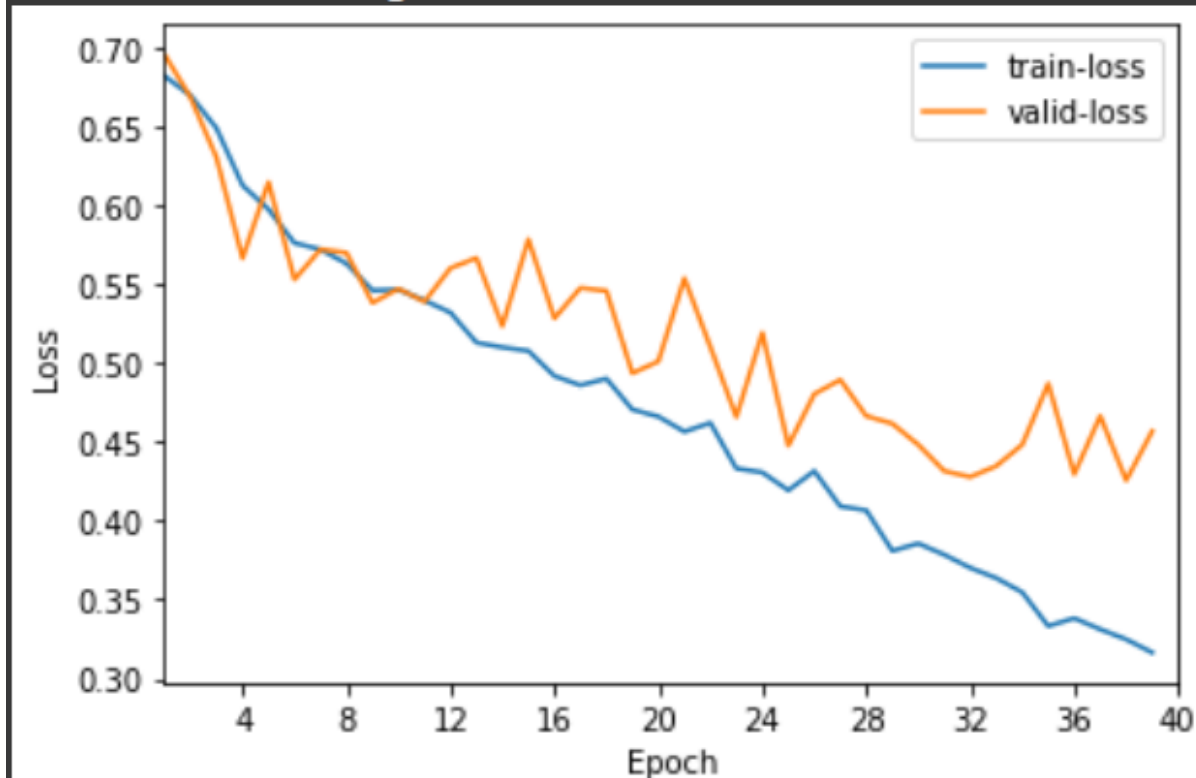
加上第四層self.cnn4, in\_channel=32、out\_channel=16、kernel\_size=2、stride=1、padding=1, 藉此強化訓練的效果, 有嘗試加上第五層, 但發現訓練效果較只有四層時差, 推斷可能是過度訓練所導致。

並且在此層做relu activation, 命名為self.relu4, 做max\_pool命名為self.maxpool4(kernel\_size=2)。

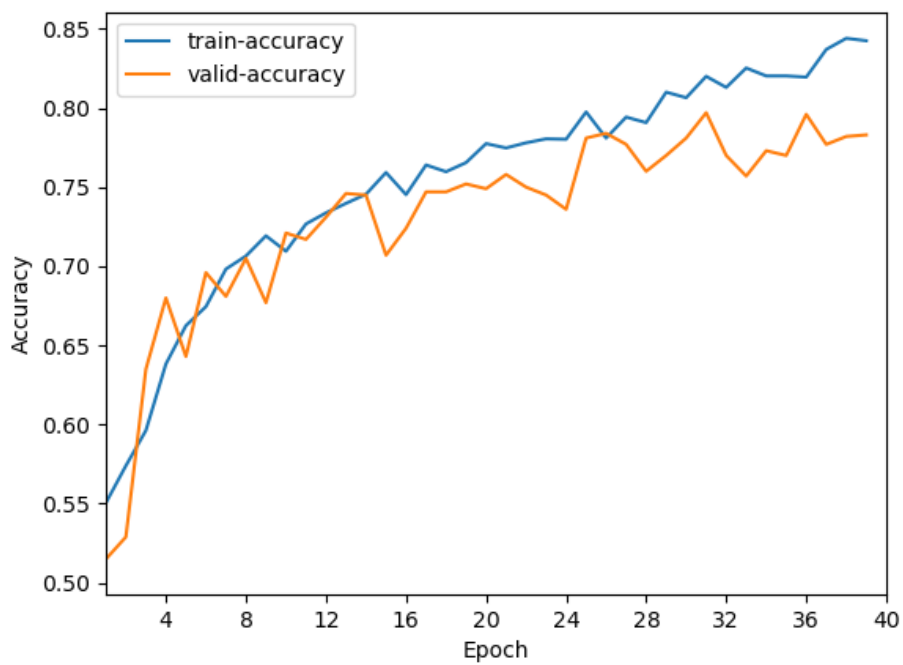
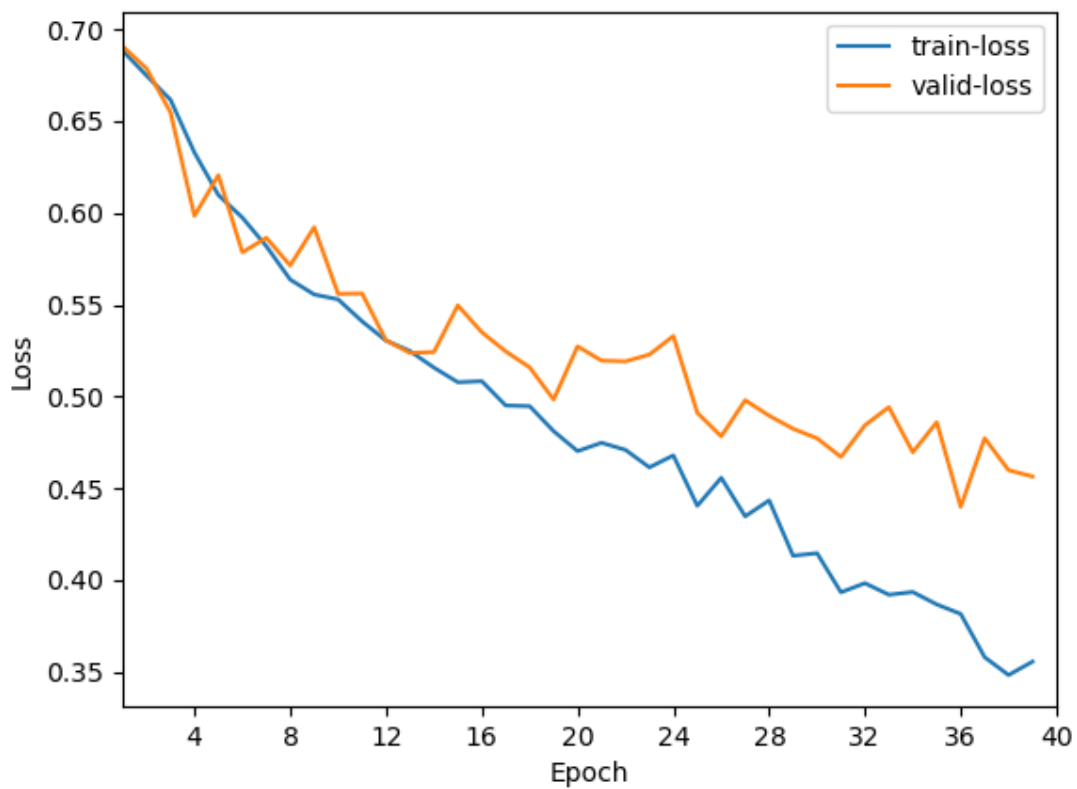
self.fc1的nn.Linear參數從原本的(64\*28\*28)改成(16\*14\*14), 64→16是因為第四層的out\_channel=16, 而28→14是因為多了一層kernel\_size=2的max\_pooling。

```
train_batch_size = 32
epochs = 40
learning_rate = 0.001
```

## Finished Training



Testing: 100% |████████████████████| 500/500 [04:20<00:00, 1.92it/s] Test accuracy: 81.4000%



Test accuracy: 80.0000%

(此為在PC上所訓練之結果，與在colab上訓練結果差不多，只差在test accuracy較低些。)

與先前的測試參數作對比

```
train_batch_size = 32
```

```
epochs = 40
```

```
learning_rate = 0.01
```

此次的model更改，在train\_loss及valid\_loss上，可發現在圖中末段，兩者的距離較近，可發現兩條曲線較契合，train\_accuracy及valid\_accuracy在末段也呈現類似的規律，兩條線彼此都相較前面數據更靠近些，代表此model訓練較先前結果緊密些，最後所產生的test accuracy為81.4%也比先前的73.9%更高些。

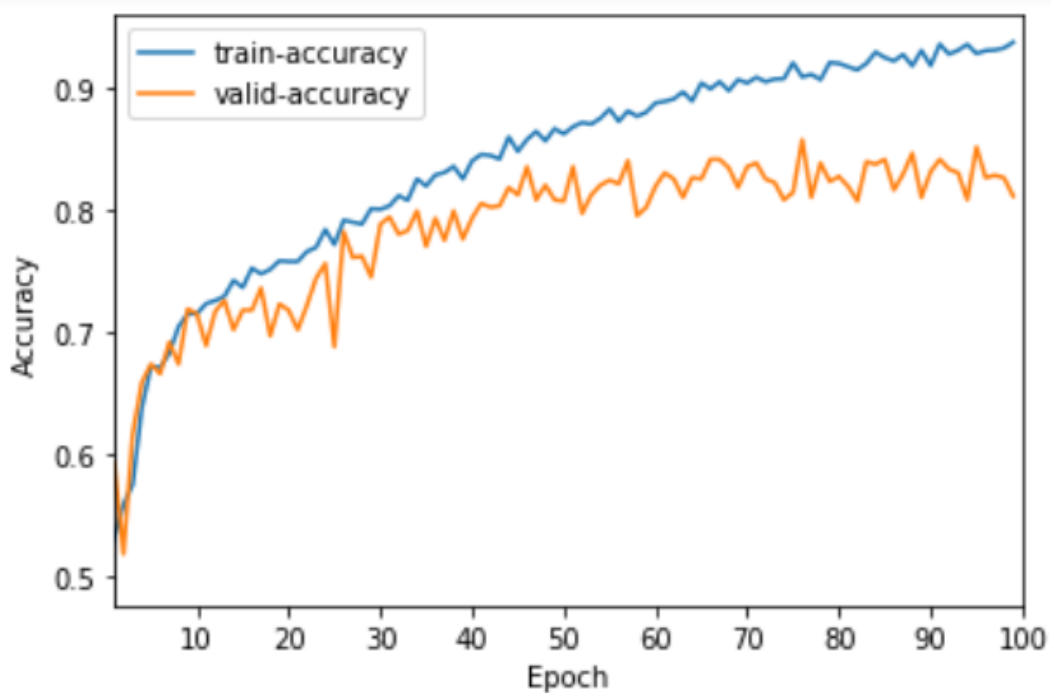
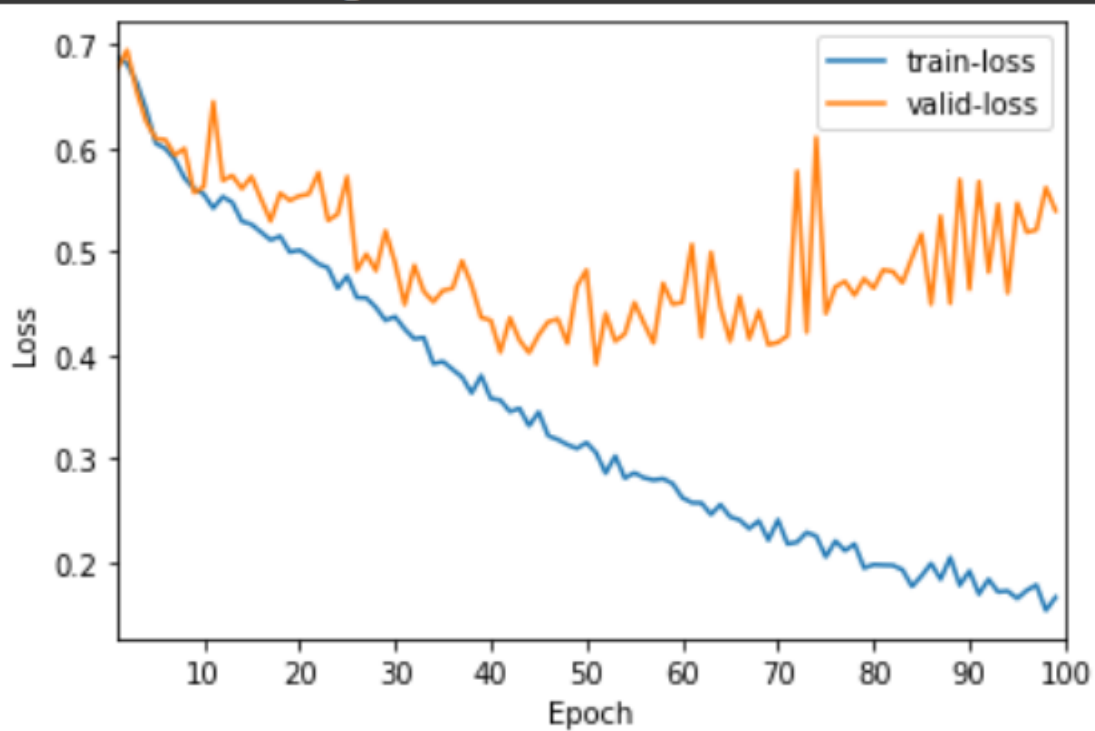
下圖所採用的參數

```
train_batch_size = 16
```

```
epochs = 100
```

```
learning_rate = 0.01
```

## Finished Training



Testing: 100% | 500/500 [00:06<00:00, 79.29it/s] Test accuracy: 82.2000%

當epoch為100次時可發現，train\_loss跟valid\_loss在epoch=40後開始，兩條線開始彼此遠離，train\_accuracy及valid\_accuracy在epoch約為45開始彼此遠離，猜測是過度訓練所導致，但由於訓練次數較多，所以train\_accuracy較高，所以最後的test accuracy也較高。