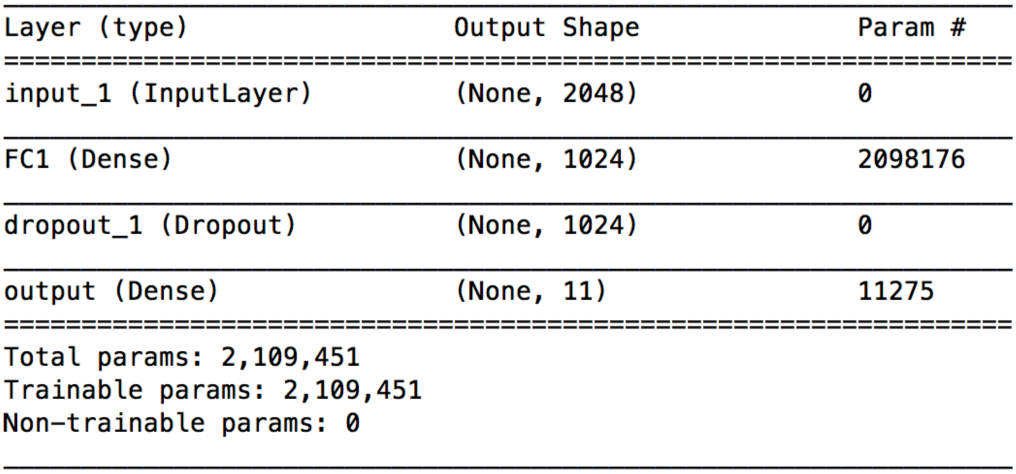
Name: 李漪莛 Dep.: 物理大四 Student ID: B03202017

[Problem1]

1. (5%) Describe your strategies of extracting CNN-based video features, training the model and other implementation details.
   - 我將所有影片利用 df (reader.py 中的參數) = 12 擷取下來，得到 x_train = (約三萬, 240, 320, 3) 的 ndim-array。
   - 我利用 Resnet50（不含 top layer）當作 base model。（試過 InceptionV3 效果很差，正確率約 2x%）
   - 將上述 x_train 通過此 base model（直接 predict，不會微調這部分的參數），得到 (~30000, 1, 1, 2048)，再通過 GlobalAveragePooling2D，得到 (~30000, 2048)，當作 features。
   - 此時，每一個影片依照時間長短會對應到不同個數的 features，我將他們直接做平均（投影片中的 strategy 1），因此每個影片此時對應到的 features = (2048,)。
   - 我建立的 CNN top layer 的結構如下：



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_1 (InputLayer)         (None, 2048)              0         
_________________________________________________________________
FC1 (Dense)                  (None, 1024)              2098176   
_________________________________________________________________
dropout_1 (Dropout)          (None, 1024)              0         
_________________________________________________________________
output (Dense)               (None, 11)                11275     
=================================================================
Total params: 2,109,451
Trainable params: 2,109,451
Non-trainable params: 0
_________________________________________________________________
```

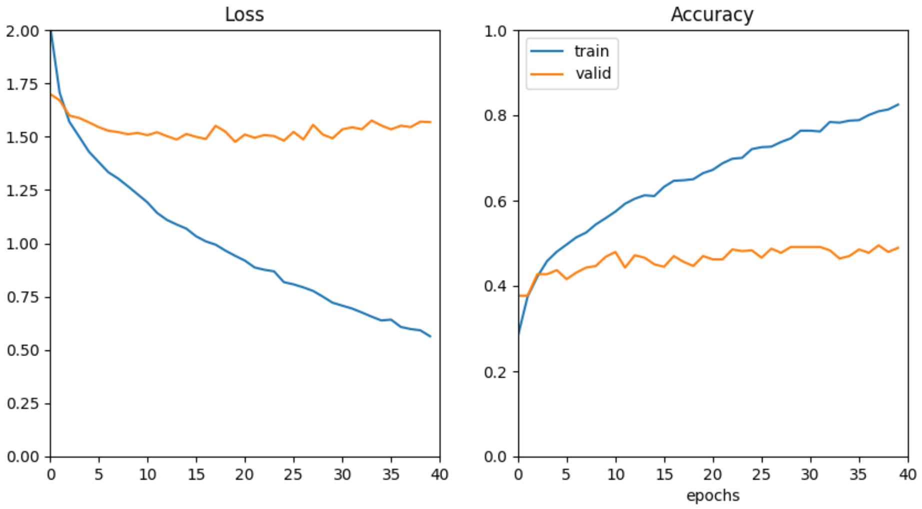
   備註：FC1 這一層有通過 relu，output 這一層通過 softmax

   - Optimizer: Adam （試過 RMSProp, SGD，效果都不太好）

     Learning rate: 1E-4

     Epochs: 40

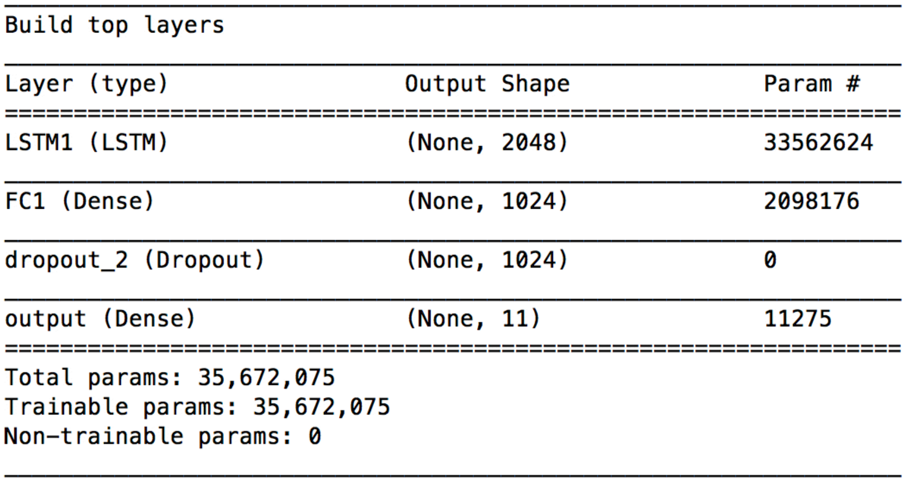
2. (15%) Report your video recognition performance using CNN-based video features and plot the learning curve of your model.
   - Accuracy: 0.48936
   - Learning curve: Blue: training data, orange: validation data



[Problem2]

1. (5%) Describe your RNN models and implementation details for action recognition.

- 使用的 features 和第一題一樣，不過在處理不同長度的影片時，我設定 max_length = 80，（training data 中有 32xx 筆資料，約只有 10~15 部影片的長度 >80），若影片長度太短，則在前面 padding zero，若長度太長，則再依固定間隔 downsample，例如：若長度為 160，則平均每兩張圖片，只取一張。此時，features_train = [None, 80, 2048]
- 我的 RNN top layer 結構如下：

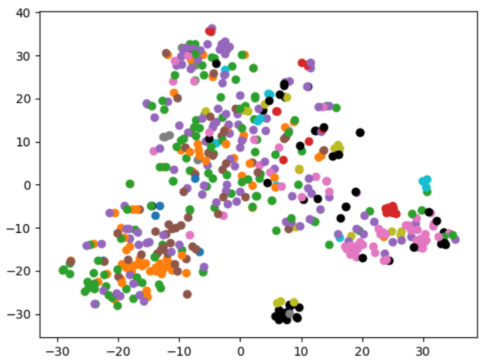
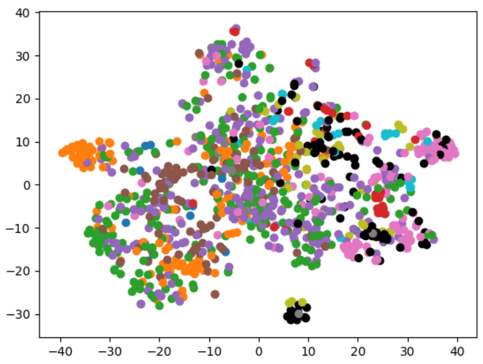


```
Build top layers
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
LSTM1 (LSTM)                 (None, 2048)              33562624
_________________________________________________________________
FC1 (Dense)                  (None, 1024)              2098176
_________________________________________________________________
dropout_2 (Dropout)          (None, 1024)              0
_________________________________________________________________
output (Dense)               (None, 11)                11275
=================================================================
Total params: 35,672,075
Trainable params: 35,672,075
Non-trainable params: 0
_________________________________________________________________
```

備註：LSTM1 那一層 dropout = 0.5，FC1 的 activation = relu，output 的 activation = softmax

- Optimizer, learning rate, epochs 同 Problem 1

2. (15%) Visualize CNN-based video features and RNN-based video features to 2D space (with tSNE). You need to generate two separate graphs and color them with respect to different action labels. Do you see any improvement for action recognition? Please explain your observation.

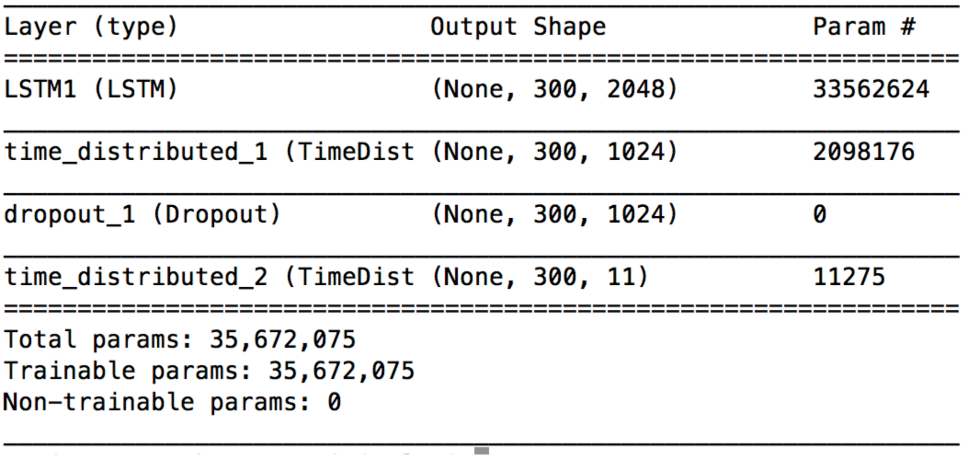
	
CNN	RNN

我覺得兩邊的 features，除了粉色和黑色點，其實都沒有分得很開，也可能是因為降到二維，導致點點看起來重疊了。

看起來有變好的，像是棕色點，CNN 的較分散，RNN 的較集中；橘色點也有發現多了一塊密集區域在左上角；其他許多顏色（如深綠，紫色）也是 CNN 的較 RNN 的更分散，RNN 的雖然沒有完全每個顏色集中成一塊，不過看得出有某個顏色較密集的區域。

[Problem3]

1. (5%) Describe any extension of your RNN models, training tricks, and post-processing techniques you used for temporal action segmentation.
   - 由於每個影片長度都很長，GPU 大小不夠，我將他們切成每段 300 張圖片，且每一片段會和前一片段有 50 張圖片的 overlap，通過 max_length = 300 的 LSTM 後，predict 好的 label 再串起來輸出。
   - 將 LSTM 後的兩層 fully connected layer 改成，可以處理每一個時間點的 LSTM 輸出（time distributed dense），就能在每個時間點都有 output，詳細架構如下：



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
LSTM1 (LSTM)                 (None, 300, 2048)         33562624
_________________________________________________________________
time_distributed_1 (TimeDist (None, 300, 1024)         2098176
_________________________________________________________________
dropout_1 (Dropout)          (None, 300, 1024)         0
_________________________________________________________________
time_distributed_2 (TimeDist (None, 300, 11)           11275
=================================================================
Total params: 35,672,075
Trainable params: 35,672,075
Non-trainable params: 0
_________________________________________________________________
```

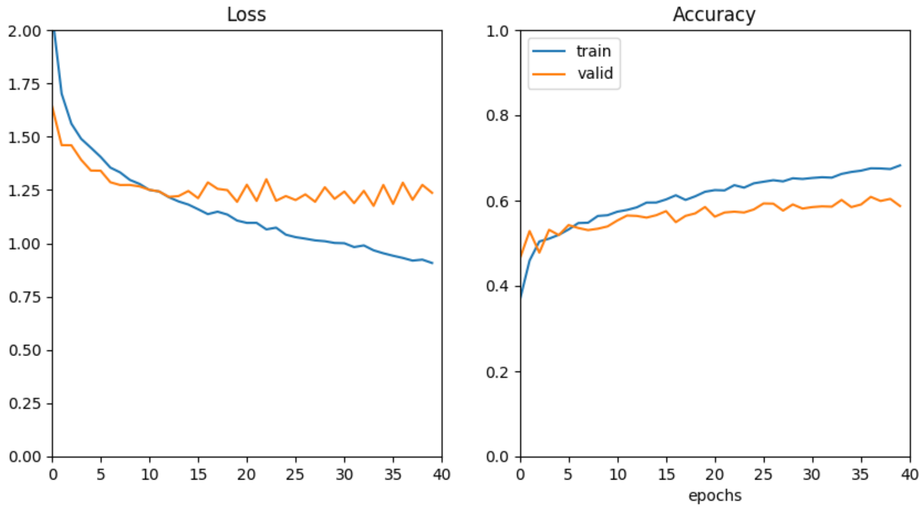
- 其他 activation, optimizer 等都和第二題相同

2. (10%) Report validation accuracy and plot the learning curve.

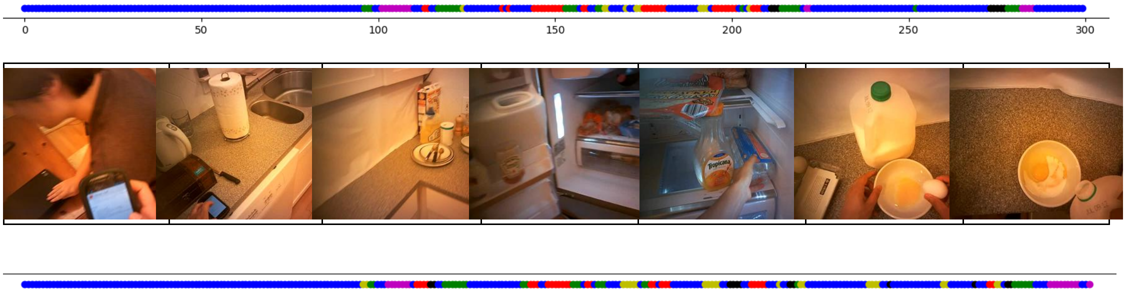
Validation accuracy: 0.631

（learning curve 圖中的正確率是包括我自己切的每段影片會有 overlapping 的部分，而在 post-processing 會去掉這些 overlapping，因此看起來會跟報告記錄的 0.631 有一點差距。）

Blue: training data, orange: validation data



3. (10%) Choose one video from the 5 validation videos to visualize the best prediction result in comparison with the ground-truth scores in your report. Please make your figure clear and explain your visualization results. You need to plot at least 300 continuous frames (2.5 mins).



- 此為 valid 第一個影片的前三百張圖片，上排是 prediction，下排是 real labels，圖片約每五十張貼上一張
- Labels 標籤：

  Blue: label 0, others

  Green: label 2, open

  Yellow: label 3, take

  Black: label 5, put

  Red: label 6, close

  M（紫色？）: label 9, pour

[BONUS]