Deep Learning on Computer Vision HW5

B04507009 電機三 何吉瑞

[Problem1]

1. Describe your strategies of extracting CNN-based video features, training the model and other implementation details.

I did the following preprocessing for this problem:

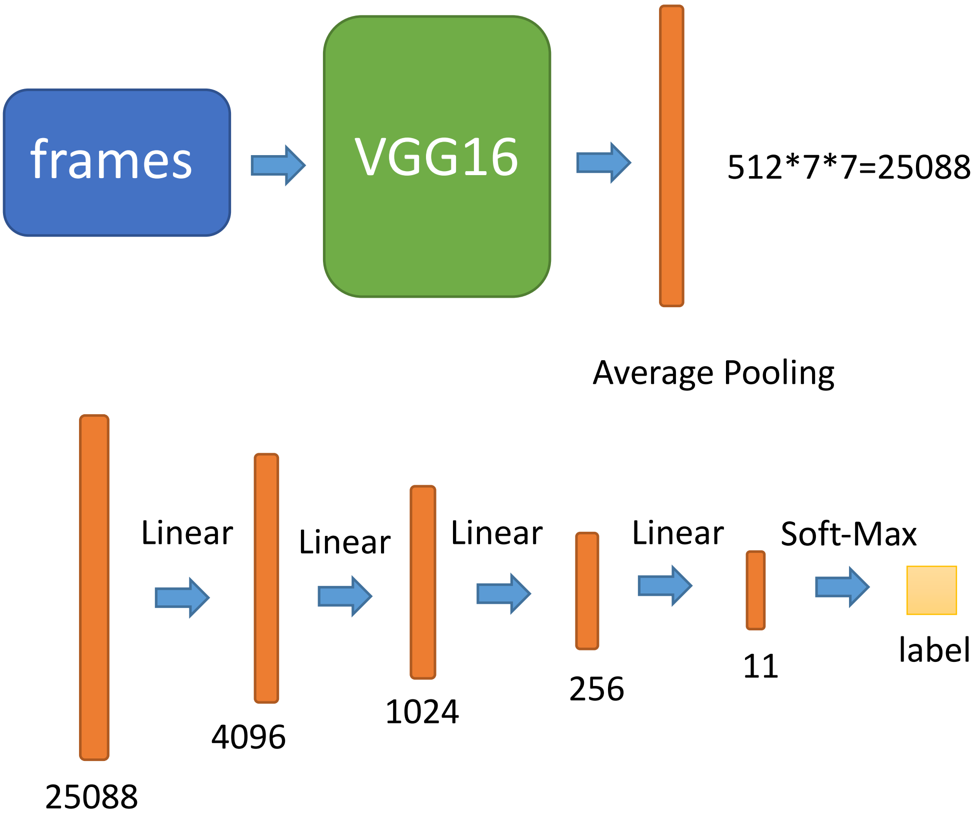
Resize and normalization: (224,224,3) with value in [-1,1]

Pre-trained CNN model: VGG16, which would turn each frame into a feature vector sized of 512*7*7 = 25088

Sample rate: 12, which is the default value

Average pooling after passing the pre-trained model for each video

The process would be like the flow chart shown below:



I apply batch-normalization for each layer.

Activation: ReLU, except the last layer(Soft-Max)

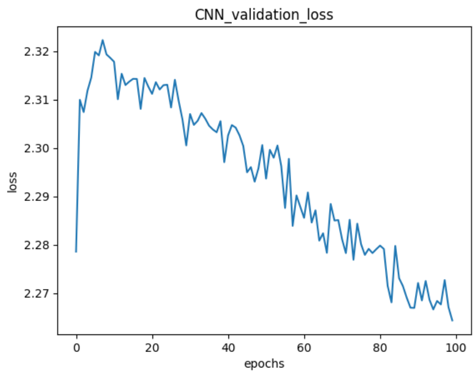
Batch size: 128.

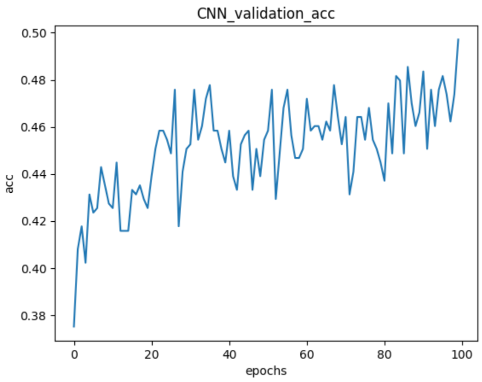
Optimizer: Adam with learning rate = 0.0001

2. Report your video recognition performance using CNN-based video features and plot the learning curve of your model

Best performance on validation set:

Loss: 2.2644 Accuracy: 49.7%



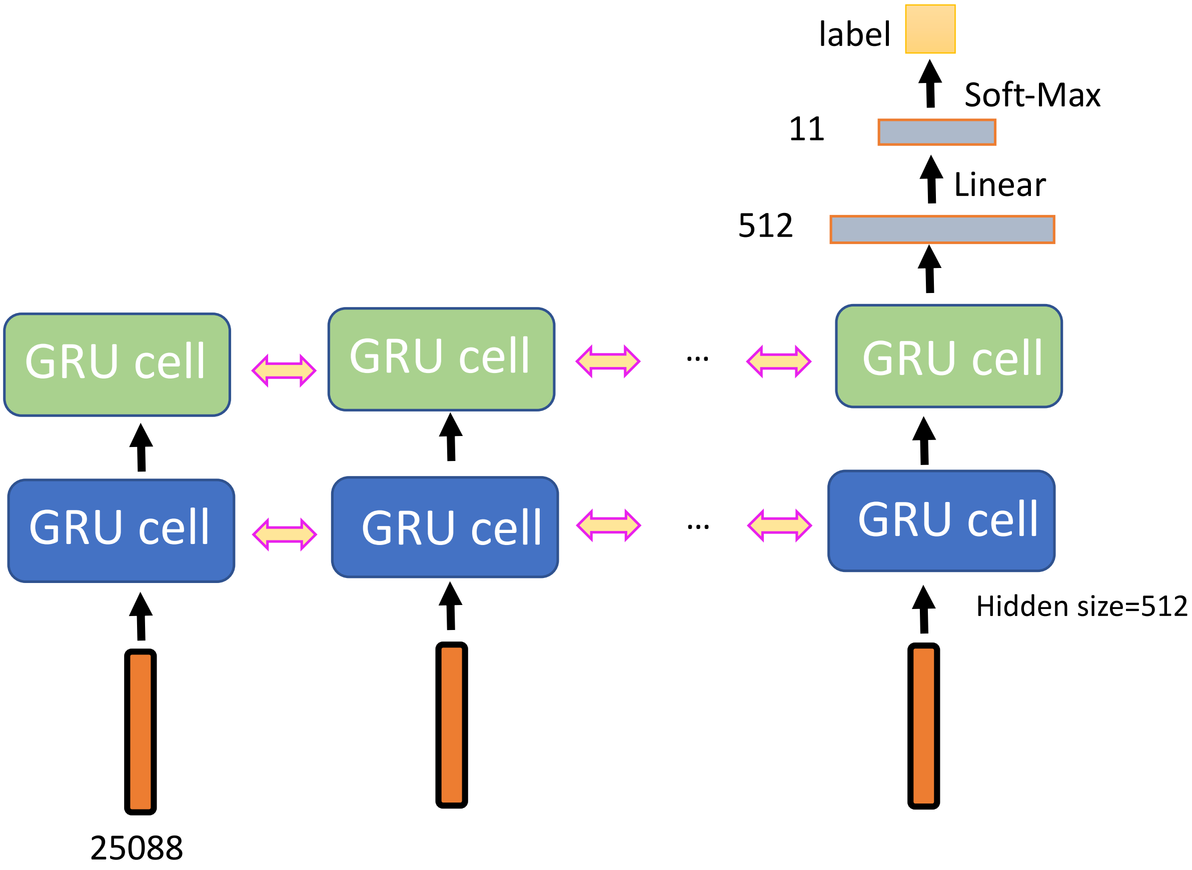


I trained for 100 epochs. It can be seen that a decreasing tendency for loss and little increasing tendency for accuracy with oscillation.

[Problem2]

1. Describe your RNN models and implementation details for action recognition

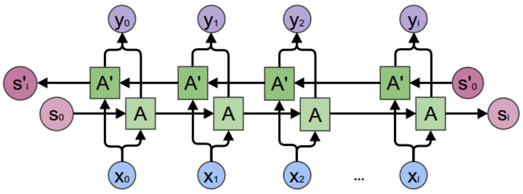
The way to generate the feature vectors sized of 25088 is as same as that mentioned in Problem1. Thus, each video would be turn into a sequence of feature vectors.



The setting of GRU cell:

bidirectional = True, layers = 2, drop out = 0.5

The above figure is simplified, the actual structure of bidirectional GRU cell is shown below.

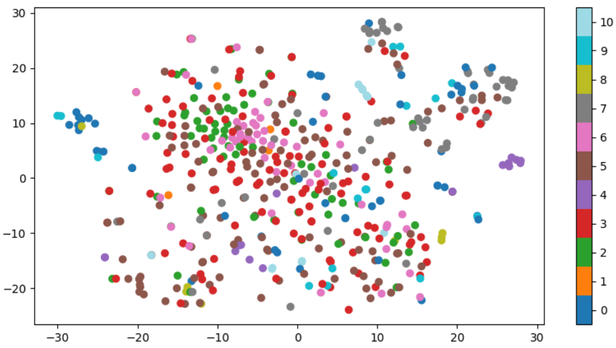


Moreover, I also used batch normalization for the linear layer.

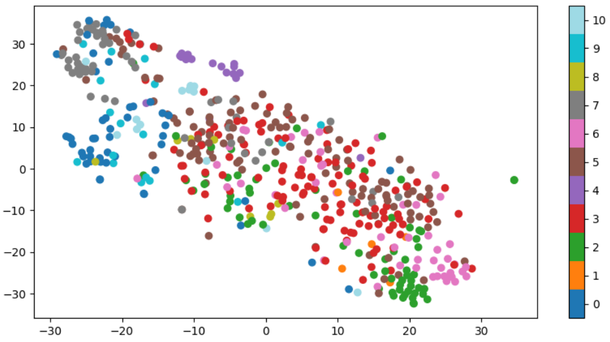
Batch size: 20, Optimizer: Adam with learning rate = 0.0001

2. Visualize CNN-based video features and RNN-based video features to 2D space (with tSNE) in your report. 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-based video features:



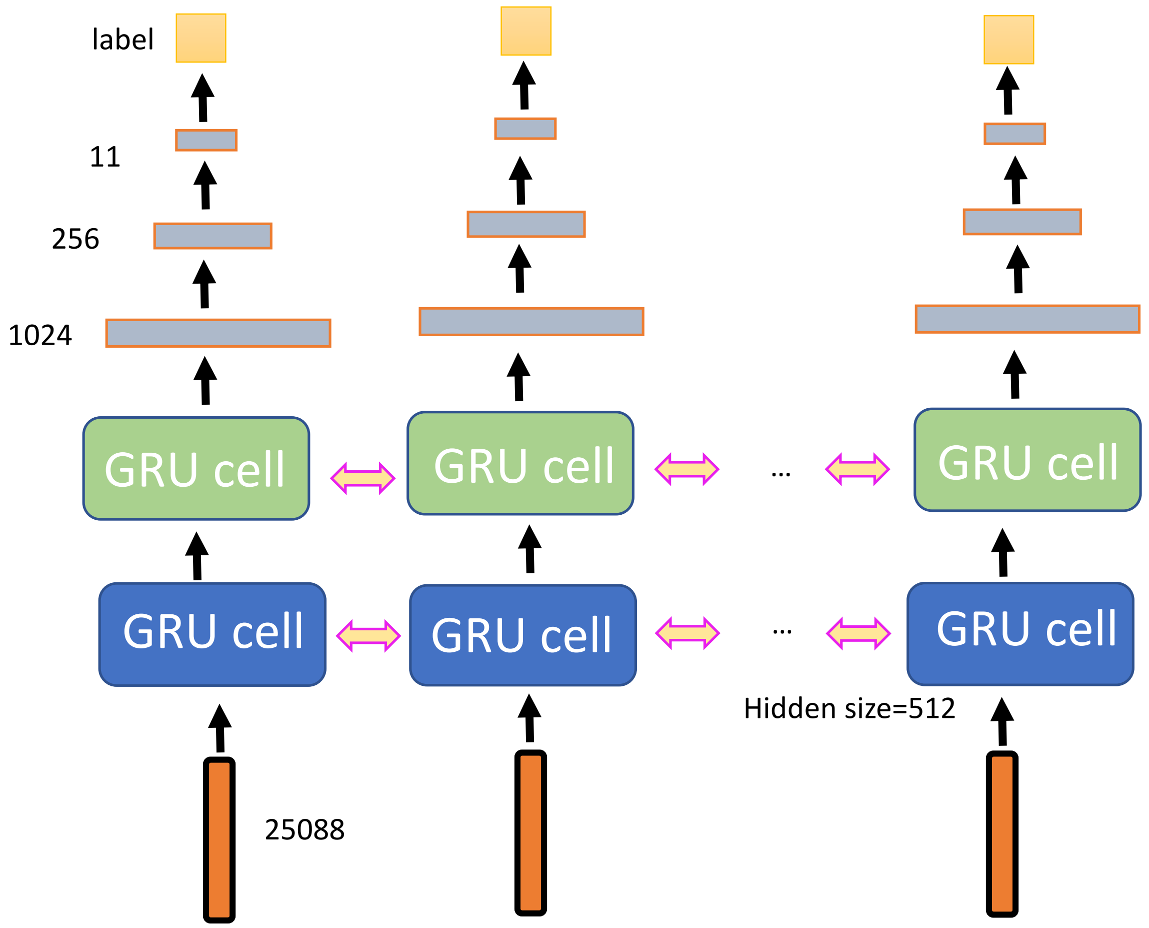
RNN-based video features:



In RNN-based video features, the clustering is more obvious, especially for class 2,4, and 7, and the pattern in CNN-based video features is difficult to discover. One reason is the difference of the original dimension of vector (CNN: 25088, RNN: 512), another reason is that for the grouping classes like 2,4, and 7, the model does learn the representation of frames, which results in better performance for RNN model.

[Problem3]

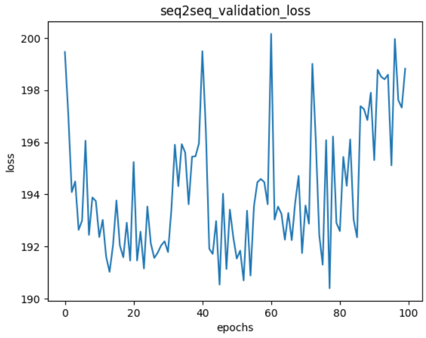
1. Describe any extension of your RNN models, training tricks, and post-processing techniques you used for temporal action segmentation.

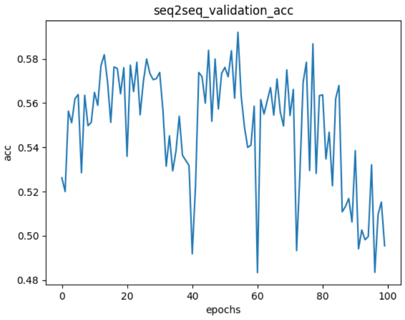


1024 = 512*2 since I used bidirectional here and get the output instead of hidden
The design of GRU is same as the previous problem.
For each video, I cut them per 64 frames in order (i.e. [0~63], [64~127], …, each is a sequence of orange vector in above figure) as the inputs of GRU layer
Moreover, I used batch normalization and it boosts the performance here.
2. Report validation accuracy and plot the learning curve (10%) in your report.
Validation accuracy:

classes	01-03	02-04	03-02	05-07	06-05
accuracy	0.657	0.563	0.539	0.487	0.605

Total: 0.5920

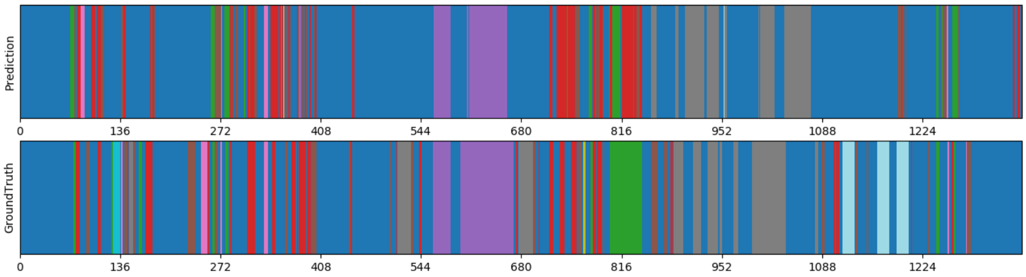




According to the learning curve, the best performance occurs during 40th ~80th epoch. After that, due to overfitting, the accuracy would be difficult to overcome the previous best one.

3. 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).

I choose OP06-R05-Cheeseburger here.
The result shows that the predictions of Other, Cut, and Move are more accurate. More specifically, it usually failed to predict labels whose time interval is short, which can be easily observed. For those labels which hold for a while, the prediction is reliable enough.


Prediction
GroundTruth