

Screenshot

39

▲ 17

Jiarui HE



0.92105

5

2d

Main Process

Conv3d

My model for the previous homework is conv2d, so I first tried to train a conv3d model which is like the previous one. But the consumption of memory is too high since it needs to store too much intermediate values for backward. I gave up this solution.

Resnet50 & LSTM

Then, I tried combining Resnet50 and LSTM to build a video classifier. To speed up the training process. I applied Adam Optimizer for the first few epochs (I tried 5,6,7 epochs) and SGD with $\text{momentum} = 0.78$ for the rest epochs. The total number of epochs is 20 and the learning rate is 4×10^{-5} . To reduce the occupation of GPU memory, the 16 frames of the middle of each video were used for training. However, this model faced the issue of overfitting, and I could not get a model with accuracy higher than 91% (it can get only $\approx 89.2\%$ in the private leaderboard).

Only Resnet50

When I tried to reduce the number of frames used for training, I noticed that the training result with few frames is usually not worse than that with more frames. The reason may be that the number of features obtained from Resnet50 is not big, and LSTM cannot get too much information from different frame since most of the frames are similar. Additionally, training with few frames can save memory and time. So, I just used the middle frame of the video (the same as the original template), applied Adam Optimizer, set learning rate as 4×10^{-5} and weight decay as default, trained for 3 epochs and got the final score.

Files:

The file structure is the same as the template in GitHub, so I will not show it here.

Reproduction

For training a Resnet50 model, the command below is used:

```
python3 CNN_Classifier_train.py resnet50 --epochs 3 --lr 0.00004 --adamEpochs 3 --weight_decay 0.1 default
```

For product the output, the command below can be used:

```
python3 CNN_Classifier_test2csv.py models/resnet50 ./output/Resnet.csv
```