Machine Learning2

Amir Jafari

Final Project - Individual Report

Changhao Ying

12/06/2020

Introduction

The project is about violence detection in video classification. With the increase in surveillance cameras throughout the world, a large amount of video data can now be collected in efforts to classify human action. For this project we will attempt to utilize the RWF-2000 video database which includes 2000 videos split evenly between violent and nonviolent actions in order to create a binary classification network to detect violence.

Each project member needs to sign up the agreement sheet (https://github.com/mchengny/RWF2000-Video-Database-for-Violence-Detection/blob/master/Agreement%20Sheet.pdf) and sent to ming.cheng@dukekunshan.edu.cn to get the dataset.

The dataset includes 2000 video surveillance clips collected from youtube sliced into 5 second clips at 30 frames per second. The database is split evenly between violence and non-violence clips.

Due to the cuda memory limitation, I picked 100 videos of violence and 100 videos of nonviolence. Preprocessing began with the cv2 package deriving 3 RGB flows and 2 Optical Flows. After creating the 5 total flows, I transformed them into npy file with shape of [nb_frames, img_height, img_width, 5].

For me, I transformed each video into 149 frame with 50 height, 50 width.

Then I assigned each video with a label (violence is 1 nonviolence is 0). I created a MLP model and a con1d model to predict the label from the npy files.

Description

Because the dataset is too large so we cannot share it on drive or github. Each member of team needs to downloaded the dataset, preprocessed and built individual models.

The preprocess step for each member is quite similar. Due to the cuda memory, I picked 100 violence videos and 100 nonviolence videos and resize the height and width into 50 , derived 3 RGB flows and 2 Optical Flows.by cv2 method. Each video is about 5 seconds or 6 seconds, for each of them, I sliced into 5 second clips at 30 frames per second. It is about 149 frames for each video. After preprocess, I got 200 npy files with (149,50,50,5) shape of violence and nonviolence and saved them on ubuntu.

I tried to pick more videos and resize into bigger one, but the memory limitation would be out. So I randomly picked 64train/16test videos to do the model.

I built two models: MLP and con1d to predict the frame is violence or nonviolence. Other team members built con2d, con3d, resent, efficient net models, etc.

For the MLP model, I firstly input all npy files I created and reshape them into (-1,1) size. So for each npy, the shape will be (149,12500). And I assigned label 1 for violence frames and 0 for nonviolence frames. I used to. Category function reshape y variable(label) into 2 number classes. I used hstack and vstack to combine all numpy array of x(frames) and y(labels) together.

To improve the accuracy of prediction, I scaled x_train and x_test sets.

```
model = Sequential()
model.add(Dense(100, input_dim=12500, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(200, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(2, activation = 'sigmoid'))
model.compile(loss='BinaryCrossentropy', optimizer=Adam(lr=0.001), metrics=['Accuracy'])
#from keras.callbacks import ModelCheckpoint
#es = EarlyStopping(monitor='val_loss',mode='min',patience=10,restore_best_weights=True)
#mc = ModelCheckpoint('mlp_cying4.h5', monitor='val_loss', mode='max',save_best_only=True)
model.fit(x_train, y_train, batch_size=512, epochs=200, validation_data=(x_test, y_test))
```

This is the best MLP model I got. I created 3 layers, more layer or less layer would decrease accuracy. For the first layer, the weight is 100 with 'relu' activation. I also dropped 0.2 proportion in the first layer. For the second layer, the weight is 200 with 'relu' activation and drop out 0.2 as well. For the final layer, the weight is 2 and activation is 'sigmoid'. The loss function is BinaryCrossentropy, optimizer is Adam with 0.001 learning rate and metrics is 'accuracy'. The batch size is 512, epochs is 200.

I tried to use SGD,RMSprop optimizer, CategoricalCrossentropy loss function, softmax, tanh, softplus, etc activations and other batch size, epochs and learning rate. But they all did not perform better compared to this one. Originally, I wanted to add an early stopping and a model checkpoint as well, but the loss was keeping decreasing and accuracy was keeping increasing through iterations so this would not be necessary.

The final accuracy is about 56.33% by model evaluation function, Cohen Kappa is about 0.56 and F1 score is about 0.12.

For the con1d model, the preprocess is quite similar with the MLP model. I also scaled each frame, and reshape each frame into(-1,1) size.

```
model = Sequential()
model.add(Conv1D(filters=3, kernel_size=5,activation='relu',input_shape=(12500,1)))
model.add(MaxPool1D(strides=3))
model.add(Flatten())
model.add(Dense(100, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(2, activation='sigmoid'))
model.compile(optimizer=Adam(lr=0.001), loss ='BinaryCrossentropy', metrics=['accuracy'])
model.fit(x_train, y_train, batch_size=200, epochs=10,validation_data=(x_test, y_test))
```

This is the best model building for con1d. Due to the cuda memory limitation, I could not increase any layers and any batch size. There are 3 layers in the model, the first layer with 3 filters, kernel size is 5, activation is 'relu' and I used maxpool1d function with 3 strides then flattened the model. For the second layer, the weight is 100 with 'relu' activation and dropped out 0.1 proportion. For the final layer, the weight is 2 with 'sigmoid' activation. The loss function is Binary Crossentropy, optimizer is Adam with 0.001 learning rate and metric is accuracy. The batch size is 100 and epoch is 10.

The final accuracy is about 53.9%, Cohen Kappa is about 0.52 and F1 score is about 0.1.

Summary

The two models I built both do not precisely predict the violence or nonviolence of each frame after scaling. The accuracy of two are not very high. It is hard for me to say which one is better.

I learned that scaling can improve accuracy a lot. For each model, I should take a try of difference activations and optimizer. Through this step, I can find which one is best.

The limitation I think is that for violence videos, there are some frames should not be considered as violence but I just assigned the violence label to each frame in the violence videos.

Improvement: I can take a try of predicting the possibility of violence or nonviolence of each frame. In this way, the frame will not be assigned violence just because it is in violence videos.

Reference

1: https://github.com/mchengny/RWF2000-Video-Database-for-Violence-Detection