

Our project will need the following environments to run:

python= 3.7

CUDA = 11.7

pytorch =1.13.0

Our AI project team provides the following interfaces:

1. External interface:

```
1. def get_train(uid: string , train_data_set:
    ndarray[n,5,56],dtype=float64) -> return (acc,logPath);
```

This function is to specialize training on a specific user's data.

uid represents the user id,

and train_data_set represents the specialization data for this user. It is a ndarray and it's size is $[n * 5 * 56]$, n means n datas, every data have 5 frames and every frame have $6 * 9$ sensor datas and one timestamps and one label.

When data error ,this funtion will throw Exception("data error")

When GPU not available,,this funtion will throw Exception("GPU error")

Return acc, which represents the training accuracy

Return logPath,to mean the where is log file path.

It is a Sample:

```
uid="zhang_asdsa"
train_data_set=np.array([[1.0 ,for x in range(1, 56)]*5])
print(get_train(uid,train_data_set))
```

The console outputs (0.92,./log/asdsad.log),mean acc=92%,and log file is in ./log/asdsad.log)

```

2. ```python
def get_predict(uid
:string,flow:ndarray[5,55],dtype=float64 ,opt :int , default)
-> return int

```

This function is to predict the current state of the user for a particular user and an array of prediction data.

uid represents the number of the user.

flow is a ndarray and it's size is $[5 * 55]$, representing the numerical values of the six sensors in 5 frames of 1 second and the current timestamps , opt=0 for calling the specialized model, and opt=1 for calling the generalization model. opt is default,you can not write it and we can decide it by function self.

For the return values: 0-6 means 7 actions, negative means an exception occurred, -1 means that the specialization model is missing and the get_train function should be called, -2 means that the data is abnormal.

It is a Sample:

```

uid="zhang_asdsa"
flow=np.array([1.0 ,for x in range(1, 56)]*5)
print(get_predict(uid,flow)) # The console outputs 0,
indicating that the prediction for this second is sitting,and
opt is default

```

```

3. def clear(uid:string): -> void

```

The purpose of this function is to clear the specialization model for that user.

uid represents the number of the user,and this function does not return a value.

It is a Sample:

```

uid="zhang_asdsa"
clear(uid) # The user specialization model is cleared

```

4.

```
def get_train_time(train_data_set:
ndarray[n,5,56],dtype=float64) : -> int #(The return time
is in seconds)
```

This function predicts the time to train. The input is the train data set and its define is same with function: get_train and the output is the estimated time to train in seconds

It is a Sample:

```
uid="zhang_asdsa"
train_data_set=np.array([[1.0 ,for x in range(1, 56)]*5])
print(get_train_time(train_data_set))
# The console outputs 10,mean need 10 seconds to train
```

5.

```
def get_progress(uid:string,train_data_set:
ndarray[n,5,56],dtype=float64): ->
(ndarray[7],dtype=float64,logPath)
```

This function show the preson data Collection progress,uid represents the user id, and train_data_set is same with function get_train.

Returns a seven-tuple representing the collection progress of each tag

Return logPath,to mean the where is log file path.

It is a Sample:

```
uid="zhang_asdsa"
train_data_set=np.array(null)
print(get_progress(uid,train_data_set))
#The console outputs ([0,0,0,0,0,0,0],./log/sad.log ,mean
every progress is 0 and log file is in ./log/sad.log
```

- 6.

```
def get_state(uid:string): -> int
```

This function show model data state.

uid represents the user id, and return a int number, return 1 means for this user has specialized model, and return 0 means for this user only have generalization model.

```
uid="zhang_asdsa"  
print(get_state(uid,train_data_set))  
#The console outputs 1,mean this user have specialized  
model.
```