INNOVATION

誰的骨盆最端正

指導教授：蘇　黎教授

指導老師：何宣螢老師

227 14 張杰、227 02 王楷睿

物理組 PHYSICS

摘要

INNOVATION

透過機器學習自動生成和音樂相符的舞蹈

訓練資料處理

訓練模型撰寫

跳舞影片

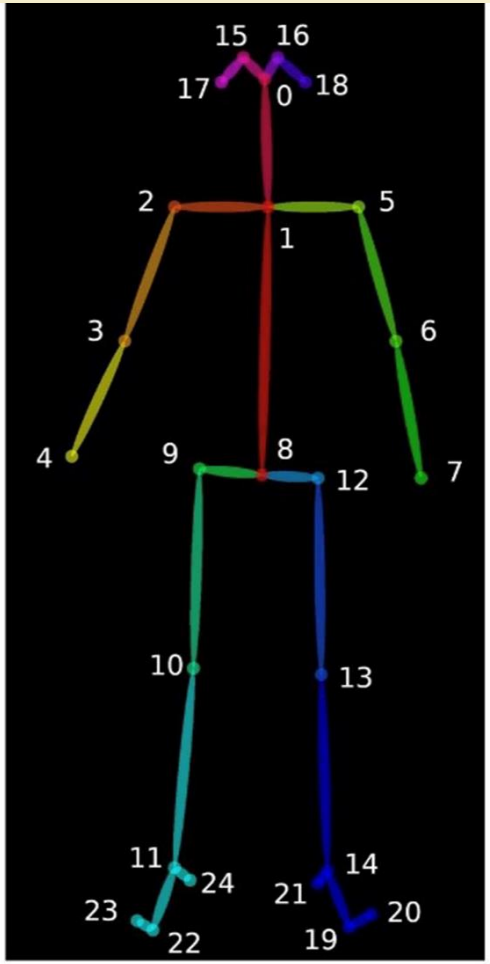
RNN(LSTM)

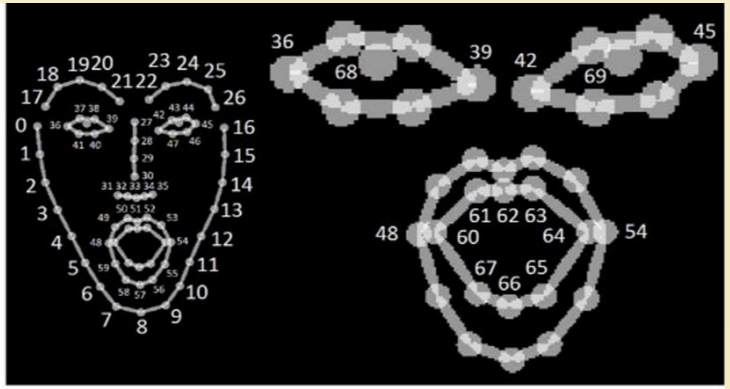
舞蹈

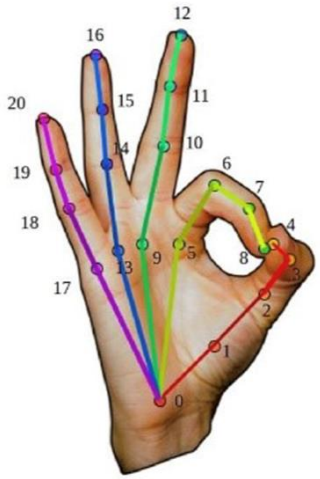
音樂

INNOVATION

Openpose







INNOVATION

輸出格式

- 第一種

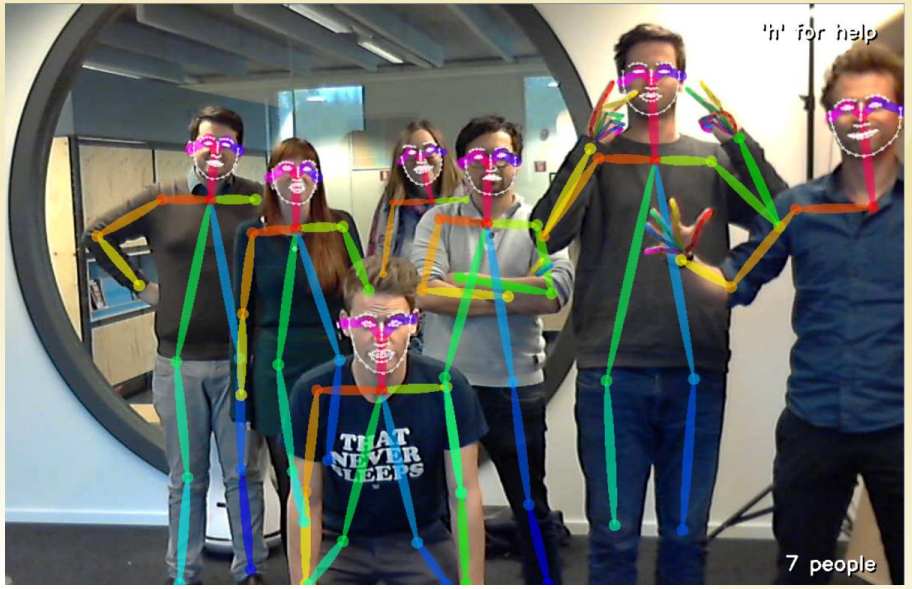
  輸入：照片、影片.....

  輸出：圖像合關鍵點的顯示(JPG檔、PNG檔)

- 第二種

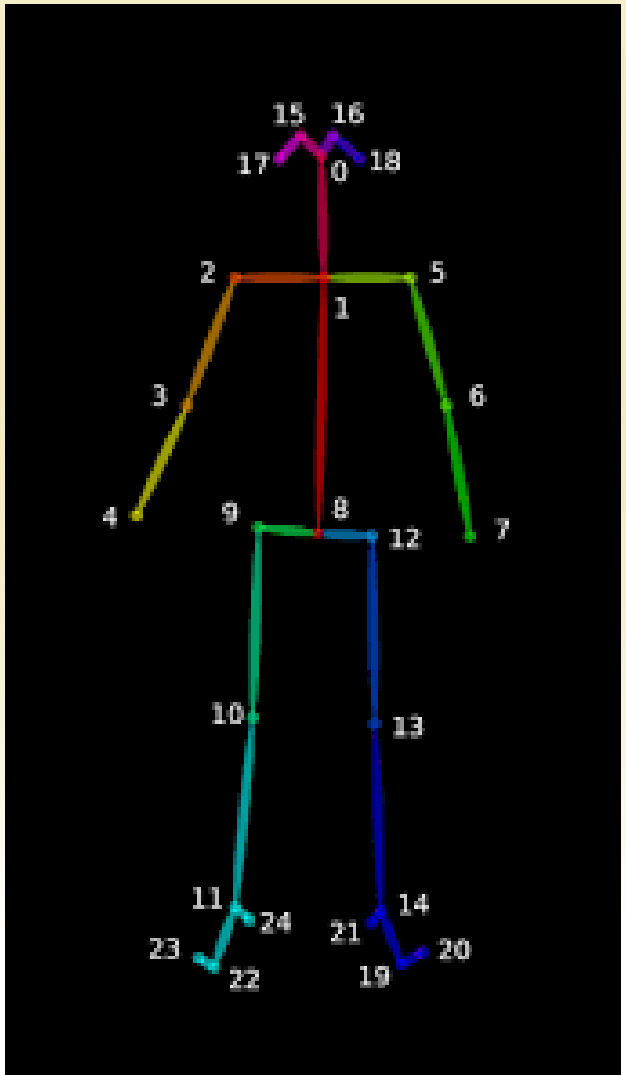
  輸入：圖片、影片.....

  輸出：關鍵點的保存

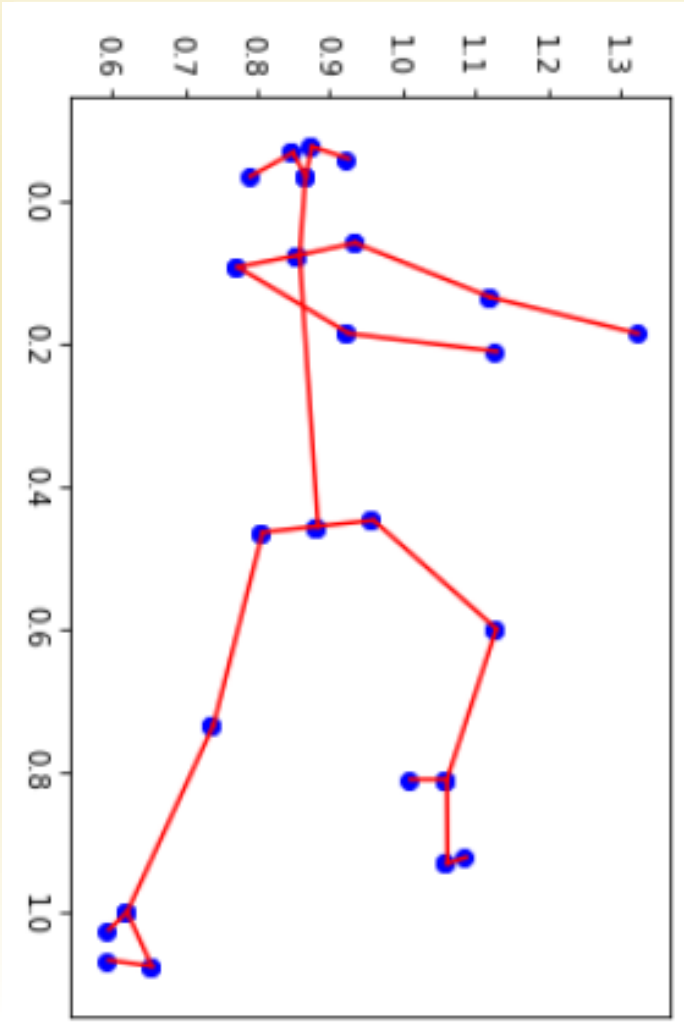


INNOVATION

輸出格式(2)



INNOVATION



INNOVATION

```
from sklearn import preprocessing
import json
import numpy as np
i=0
ar=np.ndarray((50,540))
arr=np.ndarray((50,540))
arrr=np.ndarray((3,540,50))
for x in range(1,4):
  b=str(x)
  for i in range(0,540):
    if i<=9:
      name="00"+str(i)
    elif i<=99:
      name="0"+str(i)
    else:
      name=str(i)
    with open("/content/drive/My Drive/resultjson/same"+b+"_000000000"+name+"_keypoints.json") as json_file:
      data = json.load(json_file)
    for p in data['people']:
      array=p['pose_keypoints_2d']
      a=0
      for a in range(0,25):
        ar[2*a,i]=array[3*a]
        ar[2*a+1,i]=array[3*a+1]
  arr=preprocessing.scale(ar)
  for c in range(0,50):
    for d in range(0,540):
      arrr[x-1,d,c]=arr[c,d]
```

開啟每一幀所抓的關鍵點座標

把每個座標放入相對應的位置並且標準化

訓練資料

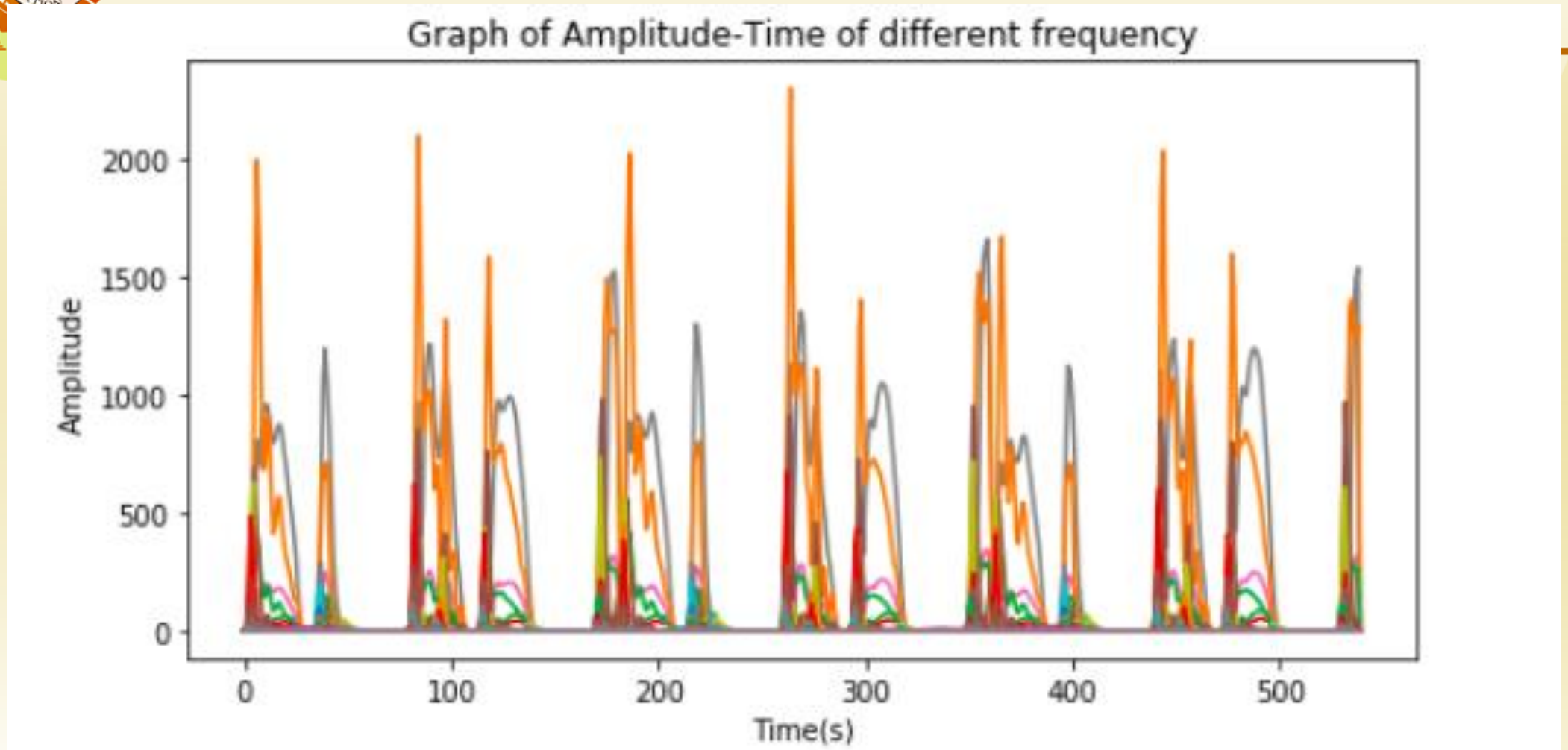
- 二維陣列的內容：

畫格 \ 座標					
第1幀第1個點的x座標	第1幀第1個點的y座標	第1幀第2個點的x座標	……	第1幀第25個點的x座標	第1幀第25個點的y座標
第2幀第1個點的x座標	第2幀第1個點的y座標	第2幀第2個點的x座標	……	第2幀第25個點的x座標	第2幀第25個點的y座標
……	……	……	……	……	……
最後一幀第1個點的x座標	最後一幀第1個點的y座標	最後一幀第2個點的x座標	……	最後一幀第25個點的x座標	最後一幀第25個點的y座標

INNOVATION

```
import numpy as np
import pandas as pd
!pip install librosa
!pip install display
import matplotlib.pyplot as plt
from glob import glob
import librosa as lb
%matplotlib inline
import seaborn
import numpy, scipy, matplotlib.pyplot as plt, IPython.display as ipd
import librosa, librosa.display

#import music data
data = "/content/drive/Shared drives/專題研究/Short music - Short instrumental - Short beat.mp3"

#get data's amplitude and frequency and
audio, sr = lb.load(data)
length = lb.get_duration(audio)
freq = lb.ifgram(audio,hop_length=100)
```

INNOVATION


Graph of Amplitude-Time of different frequency
Amplitude
2000
1500
1000
500
0
0
100
200
300
400
500
Time(s)

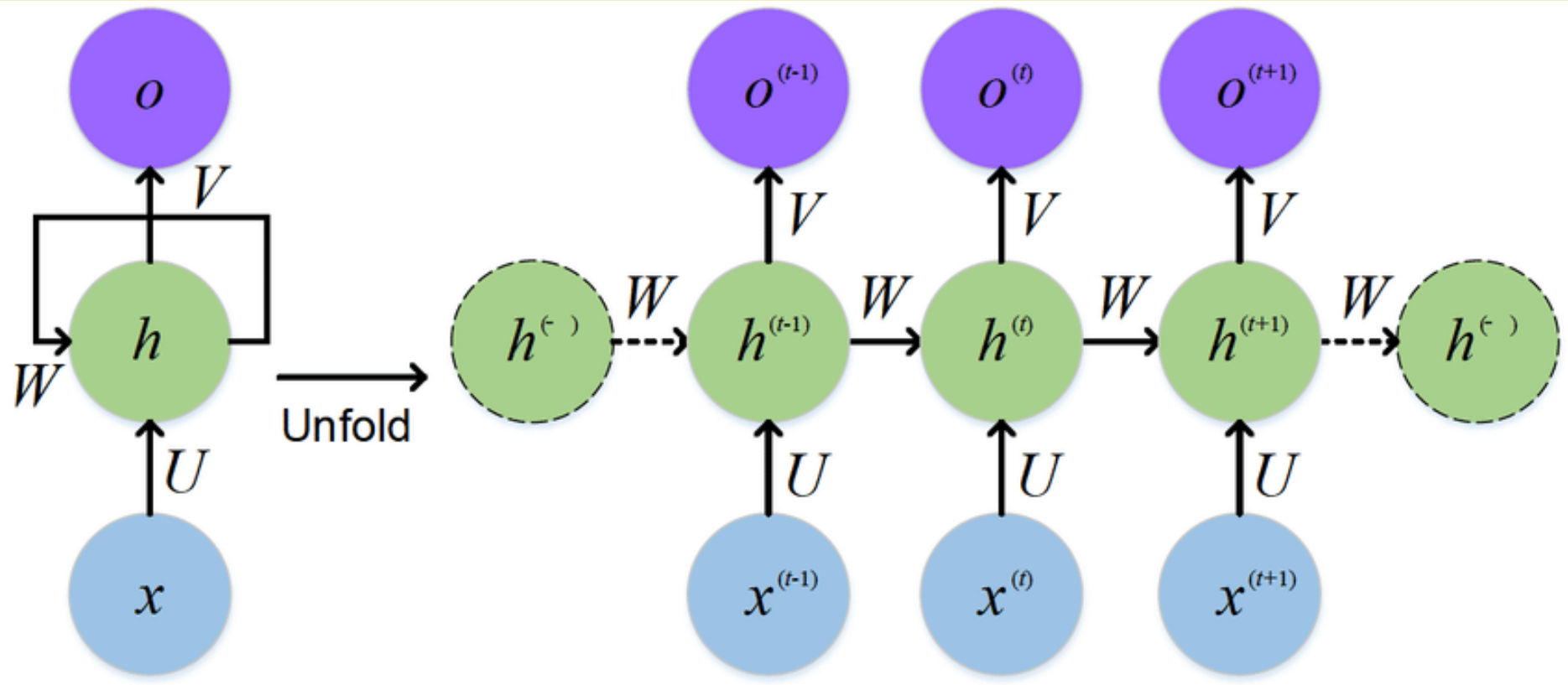
INNOVATION

Librosa 頻譜分析音樂

第一個頻率包 第一個時間點 的振幅	第二個頻率包 第一個時間點 的振幅	……	……	……	……
第一個頻率包 第二個時間點 的振幅	第二個頻率包 第二個時間點 的振幅	……	……	……	…….
……	……	……	……	……	……

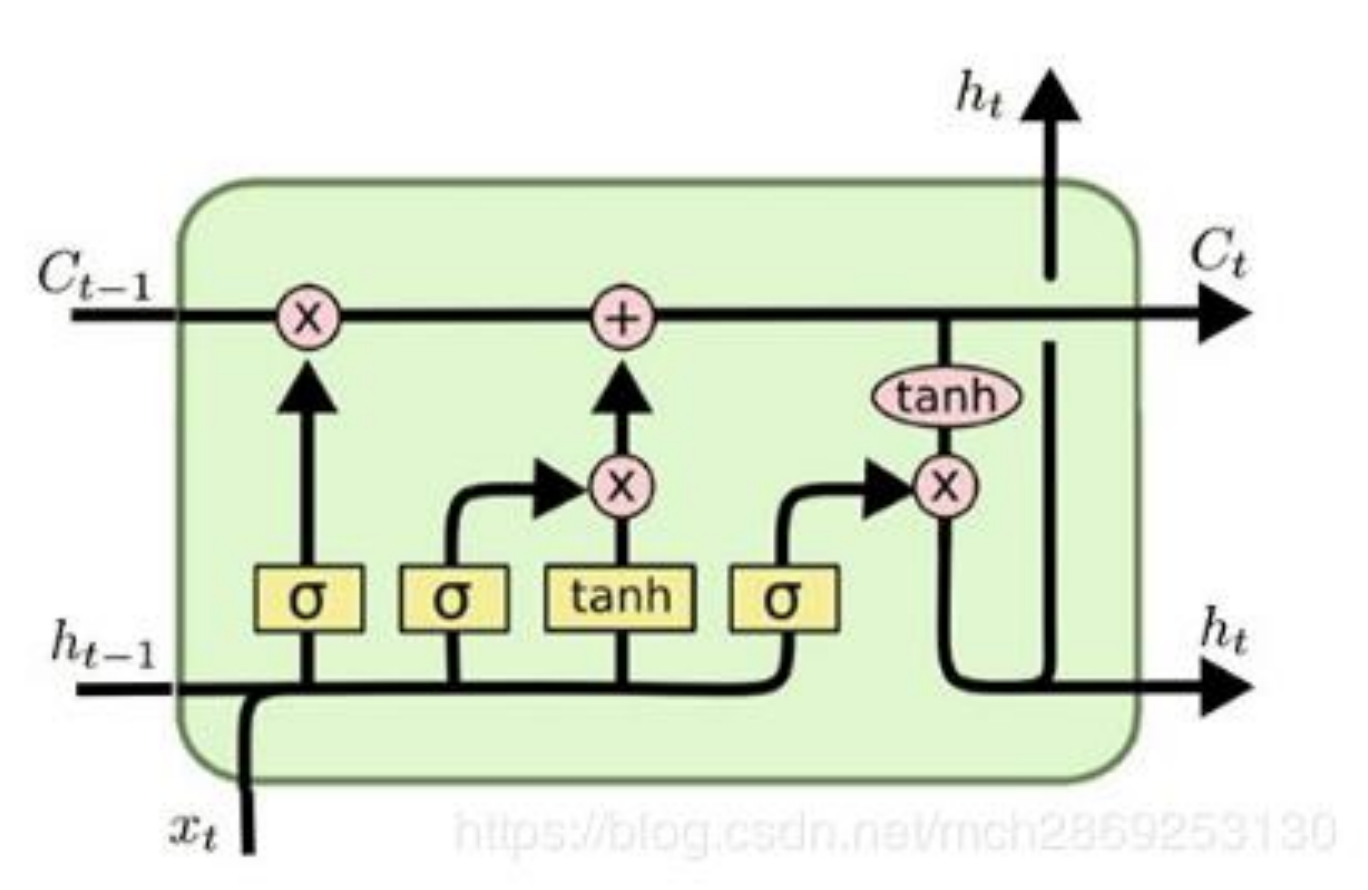
INNOVATION

Recurrent Neural Network



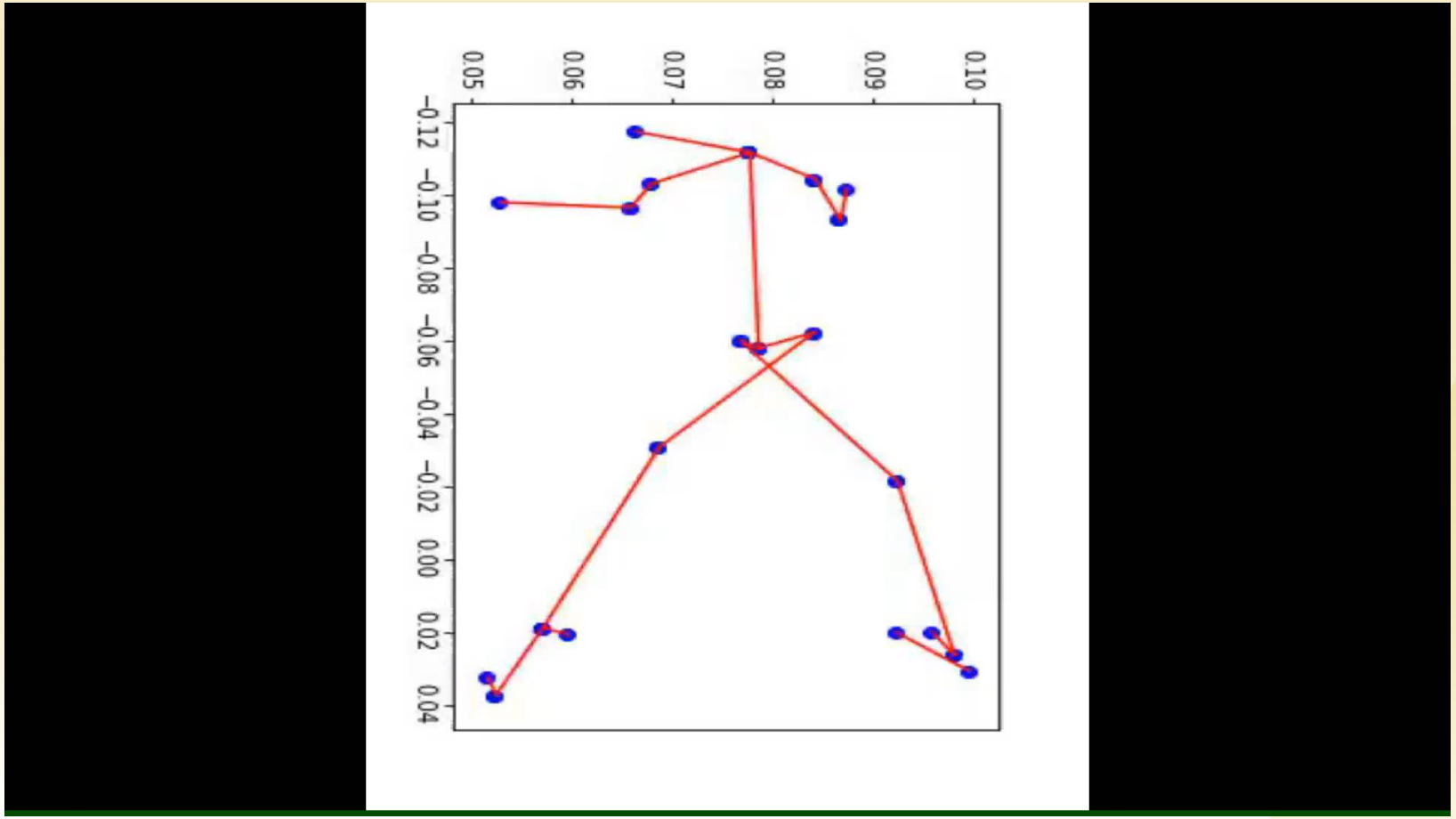
INNOVATION

Long short-term memory



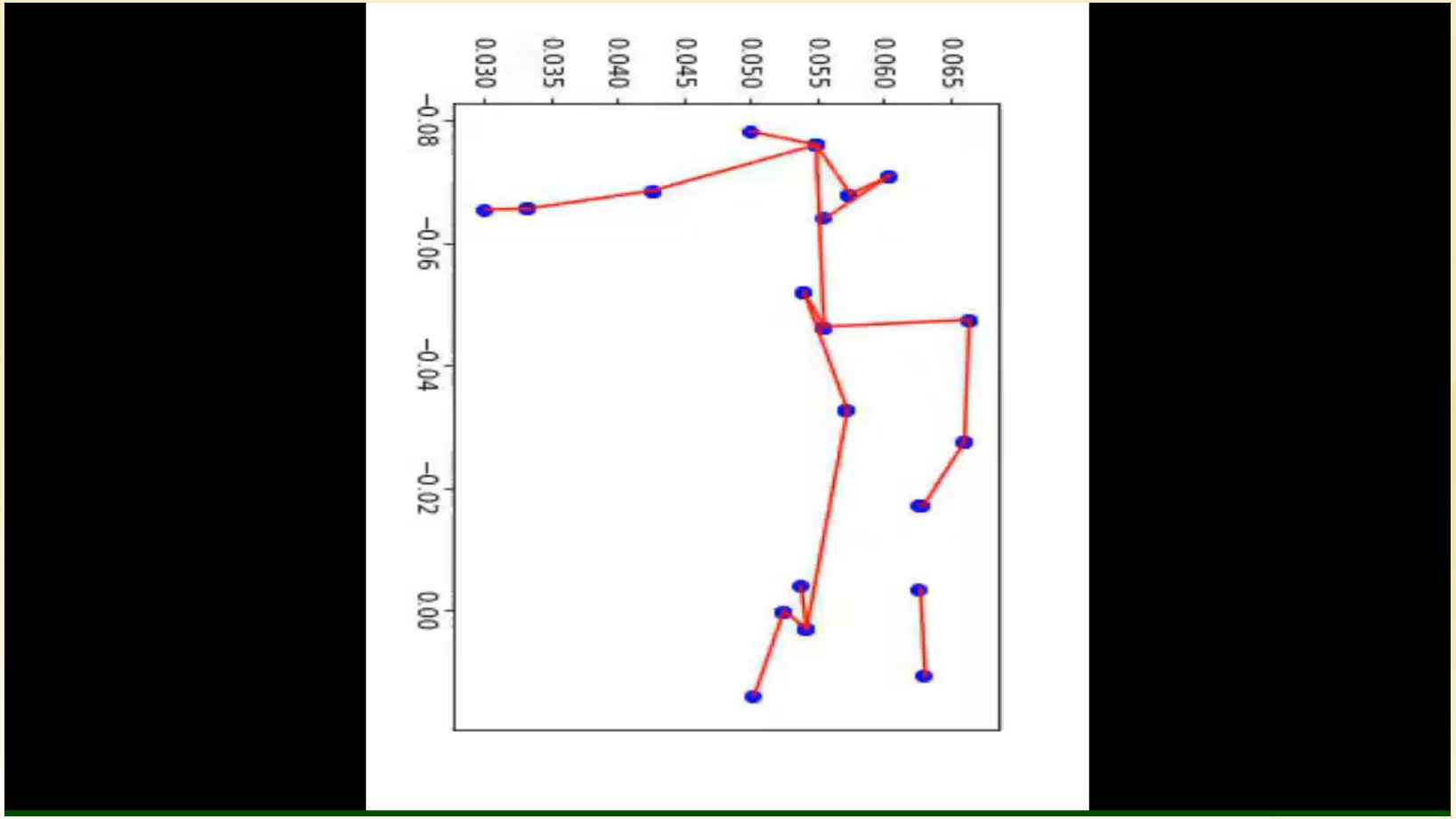

```
[ ]  # constructing model
     from keras.models import Sequential
     from keras.layers import Dense, Dropout, GRU
     from keras.layers import Embedding
     from keras.layers import LSTM,TimeDistributed
     rate=0.25

     RNN=Sequential()
     # RNN.add(TimeDistributed(model,input_shape=(540,128)))
     RNN.add(LSTM(256,return_sequences=True,input_shape=(540,128)))
     RNN.add(Dropout(0.3))
     RNN.add(LSTM(256,return_sequences=True))
     RNN.add(Dropout(0.3))
     RNN.add(LSTM(256,return_sequences=True))
     RNN.add(Dropout(0.3))
     RNN.add(TimeDistributed(Dense(50)))
     RNN.compile(loss=weighted_mean_squared_error,optimizer='Adam',)
     #RNN.compile(loss='mse',optimizer='Adam',)
     print(RNN.output_shape)
```

結果



INNOVATION

結果



INNOVATION

POWER MOVE評分機制

INNOVATION

克服的困難

- 體型影響分析（身高 比例……）
- 速度影響分析
- 肢體遮擋問題

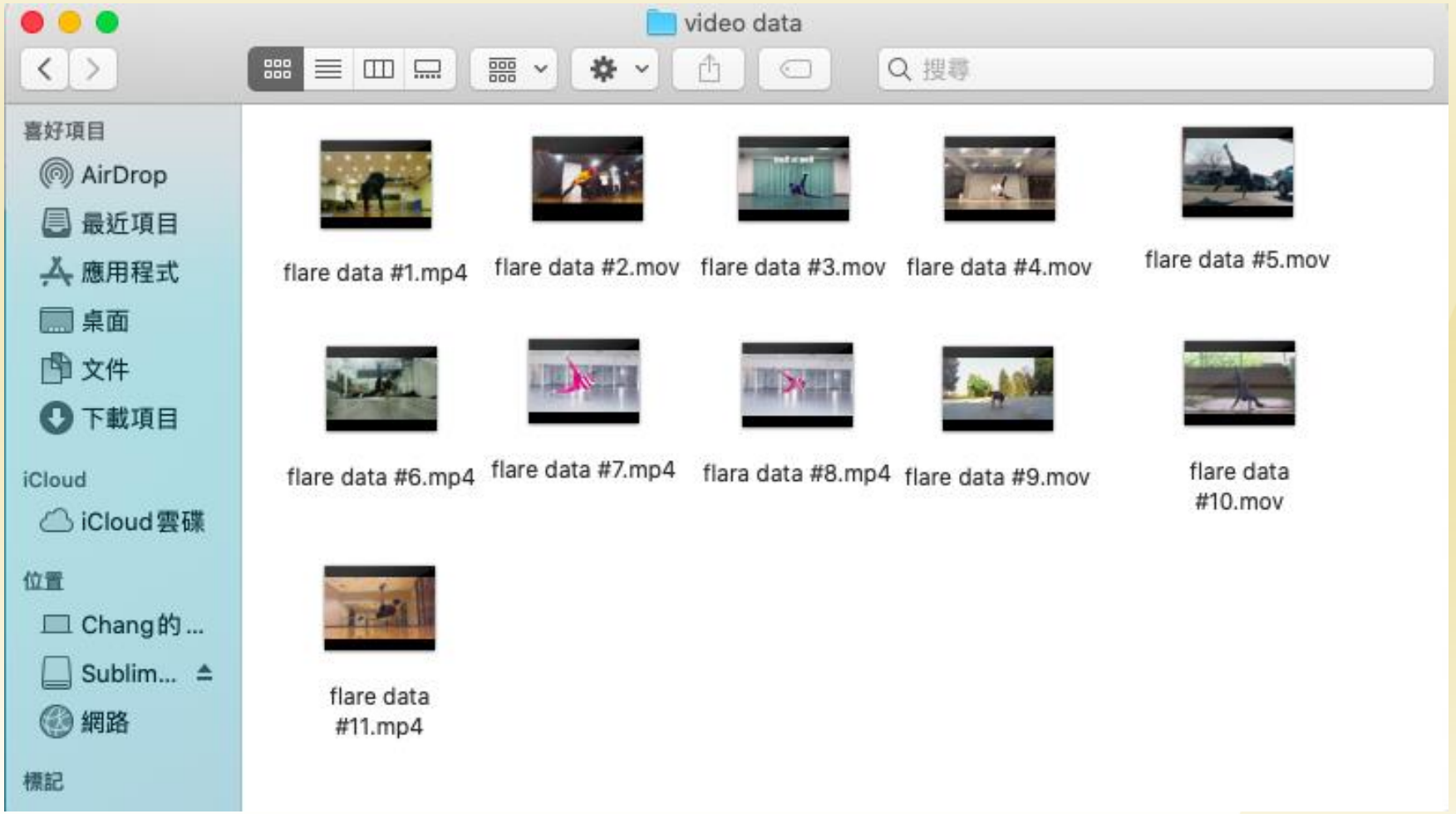
INNOVATION

流程

目前進度

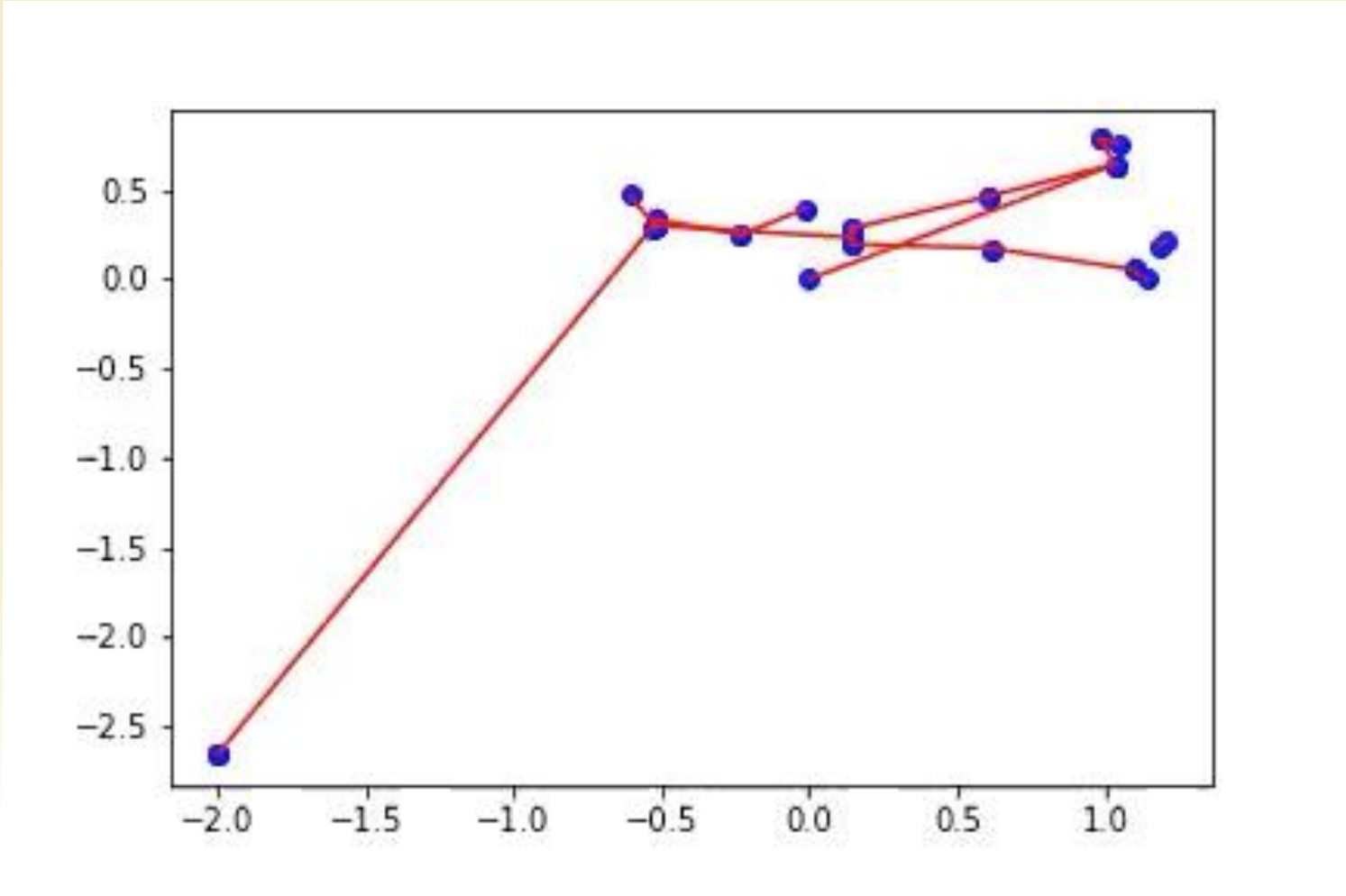
INNOVATION

收集資料



INNOVATION

Openpose讀取資料的困難

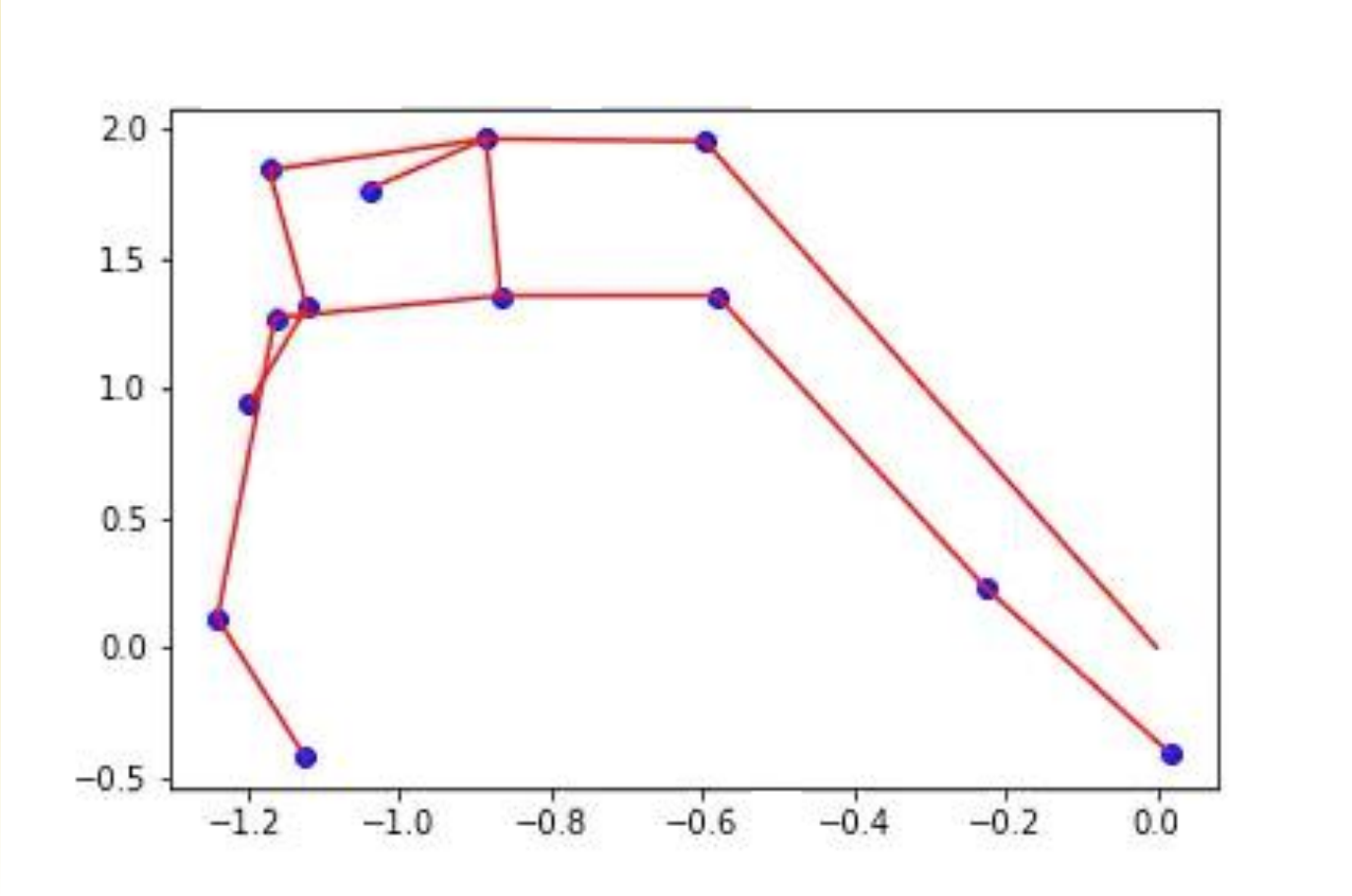


標準化過程

```
#standardization function
import math

def average(lst):
  average = np.sum(lst)/np.count_nonzero(lst)
  return average

def nd(lst):
  g=0
  for i in lst:
    if i!=0:
      g+=(i-average(lst))**2
  return math.sqrt(g/np.count_nonzero(lst))

def standardize(lst):
  a = np.ndarray((len(lst),len(lst[0])))
  for i in range(0,len(lst)):
    for j in range(0,len(lst[0])):
      if lst[i][j]==0:
        a[i][j]=0
      else:
        a[i][j]=(lst[i][j]-average(lst[i]))/nd(lst[i])
  return a
```

INNOVATION

修改標準化過程



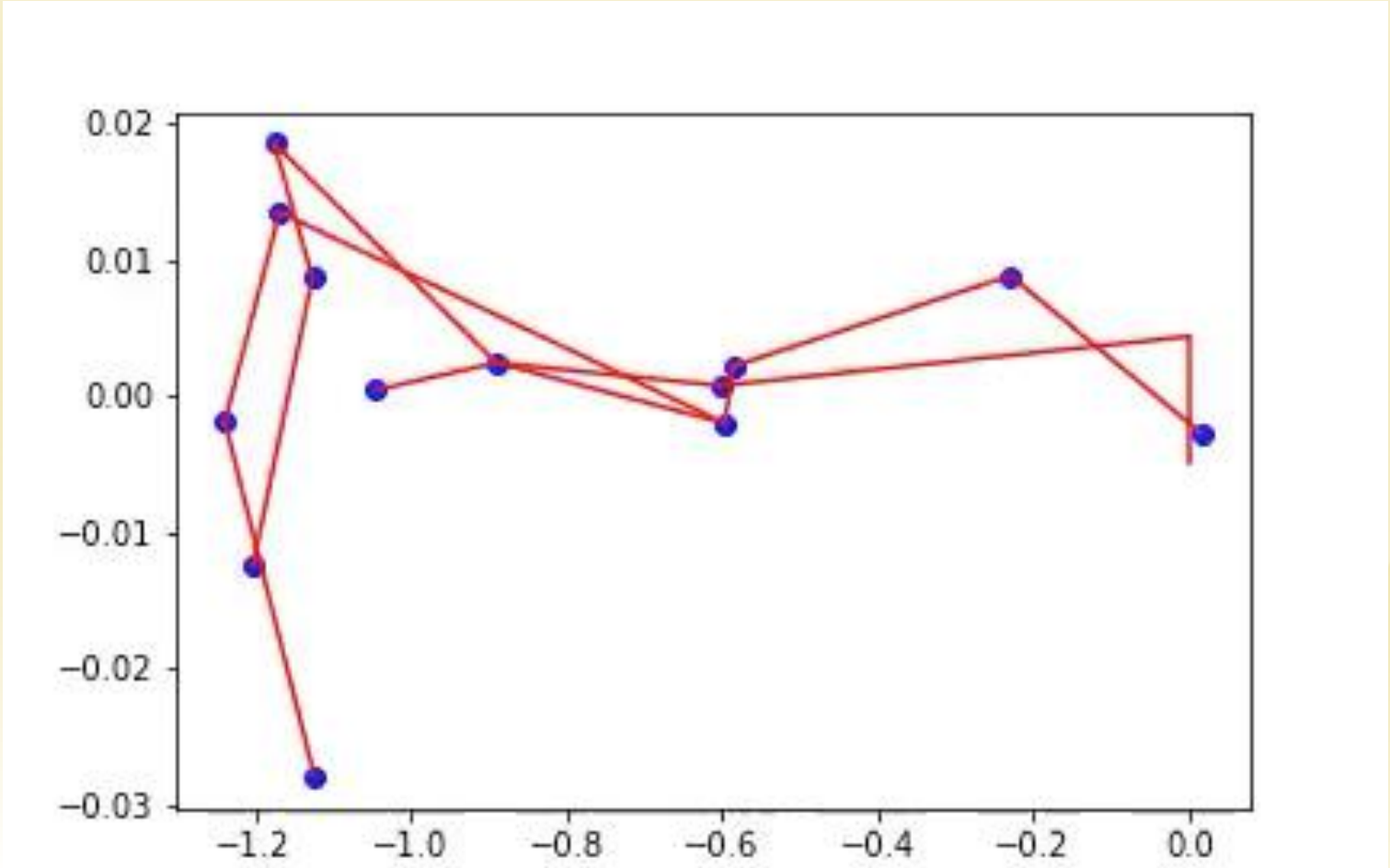
INNOVATION

Model

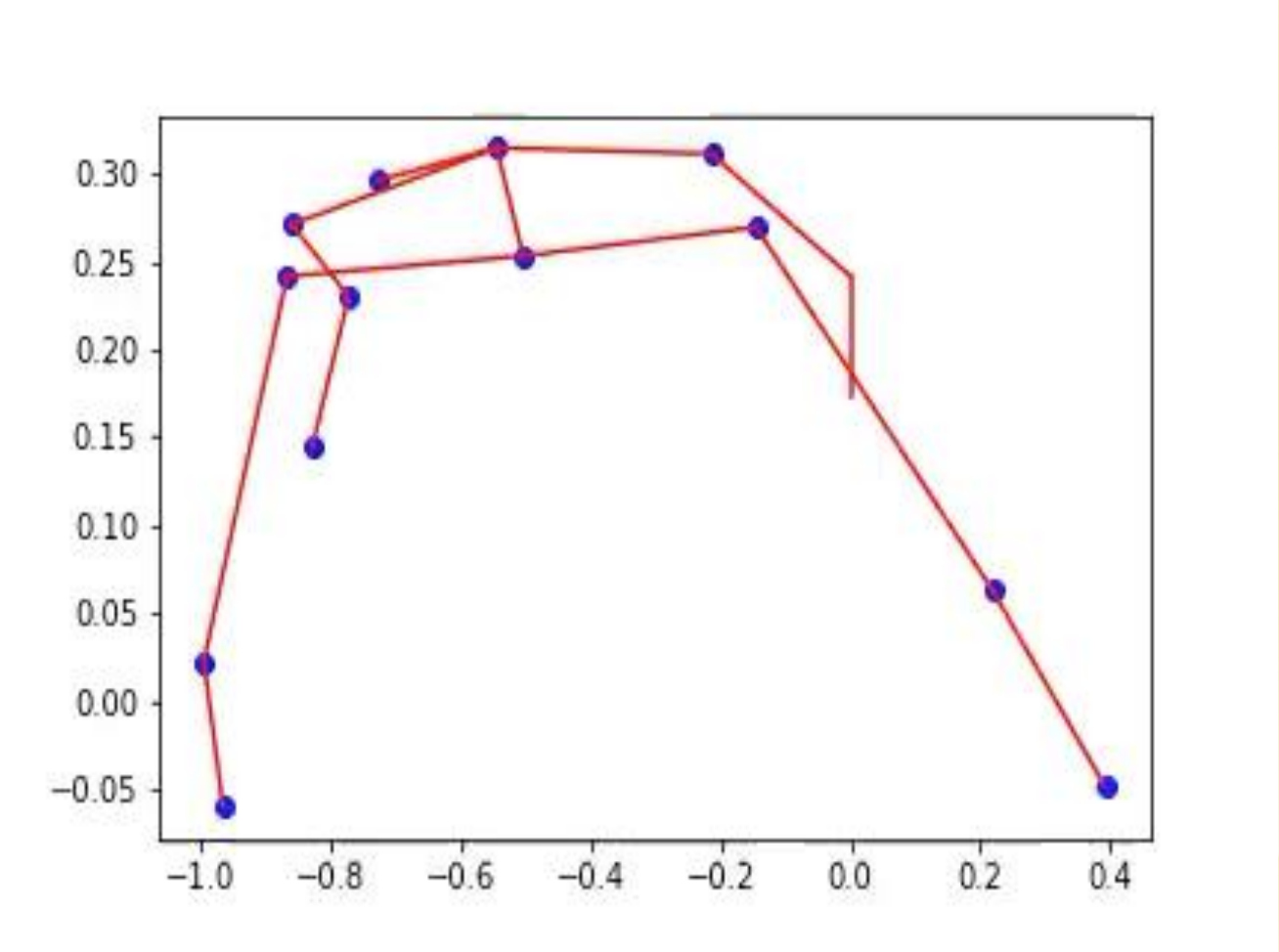
```
# constructing model
from keras.models import Sequential
from keras.layers import Dense, Dropout, GRU
from keras.layers import Embedding
from keras.layers import LSTM,TimeDistributed,Masking

rate=0.25

RNN=Sequential()
# RNN.add(TimeDistributed(model,input_shape=(540,128)))
RNN.add(Masking(mask_value=0.,input_shape=(130, 15)))
RNN.add(LSTM(256,return_sequences=True))
RNN.add(Dropout(0.3))
RNN.add(LSTM(256,return_sequences=True))
RNN.add(Dropout(0.3))
RNN.add(LSTM(256,return_sequences=True))
RNN.add(Dropout(0.3))
RNN.add(TimeDistributed(Dense(15)))
RNN.compile(loss=weighted_mean_squared_error,optimizer='Adam',)
#RNN.compile(loss='mse',optimizer='Adam',)
print(RNN.output_shape)
```

INNOVATION

Power Move 結果#1（修改標準化過程前）



Power Move 結果#2



INNOVATION

應用

INNOVATION

特別感謝

- 建國中學何宣螢老師
- 中央研究院蘇黎教授
- 中央研究院高炫凱學長

INNOVATION