Apply neural network to Human Activity Recognition

1. Background

1.1 introduction

Human activity recognition research has traditionally focused on discriminating between different activities. The goal of activity recognition is to recognize common human activities in real life settings. Accurate activity recognition is challenging because human activity is complex and highly diverse. Several probability-based algorithms have been used to build activity models.

There are many potential applications for HAR, like: elderly monitoring, life log systems for monitoring energy expenditure and for supporting weight-loss programs, and digital assistants for weight lifting exercises.

2. Model

2.1 RNN (LSTM and Bi-directional LSTM)

Compared to a classical approach, using a Recurrent Neural Networks (RNN) with Long Short-Term Memory cells (LSTMs) require no or almost no feature engineering. Data can be fed directly into the neural network who acts like a black box, modeling the problem correctly. Other research on the activity recognition dataset used mostly use a big amount of feature engineering, which is rather a signal processing approach combined with classical data science techniques.

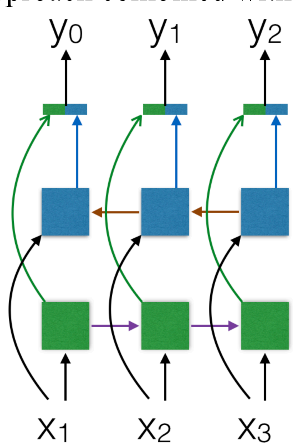

fig2.1 bidirectional LSTM

I want to only discuss about bi-directional LSTM here(We have learnt LSTM before). The figure above shows the structure of the Bidirectional LSTM, whereby you have one forward LSTM and one backward LSTM running in reverse time and with their features concatenated at the output layer, thus enabling information from both past and future to come together.

2.3 CNN+ RNN (LSTM/ Bi-directional LSTM)

According to a blog, Convolutional Neural Network (CNN) can be apply for HAR, that will learn complex features automatically from the raw accelerometer signal to differentiate between different activities of daily life.

The figure below provides the CNN model architecture that we are going to implement using Keras. The model will consist of one convolution layer followed by max pooling and another convolution layer following by max pooling. Another important thing is that the convolution and max-pool layers will be 1D or temporal.

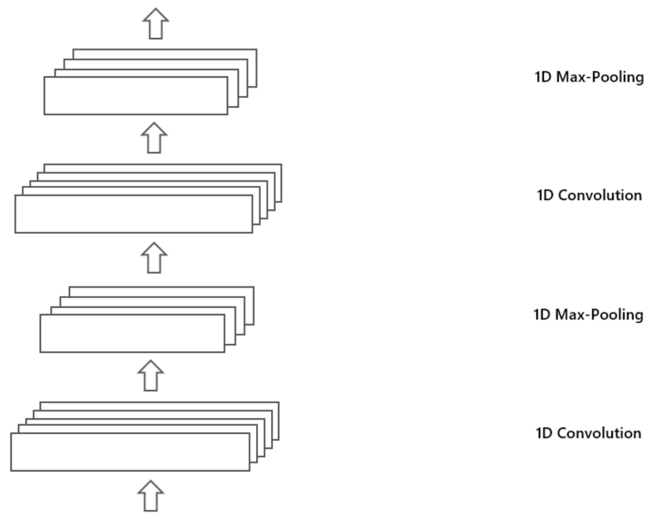

fig 2.2 my CNN model

In the future, I want to change my CNN model as a more complete model.

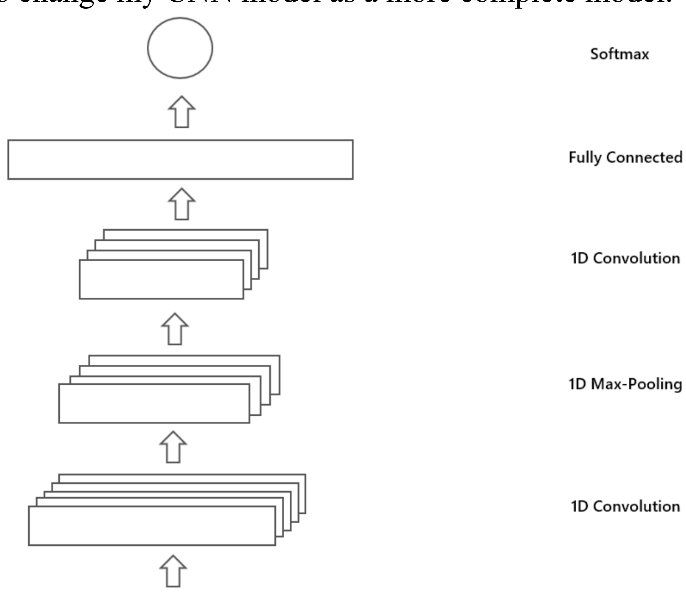

fig2.3 new CNN model

Because of we use 1-D convolutional neural network, there is no dimension problem. Therefore, we do not need to pay more attention when we connect CNN and RNN.

3. Result

3.1 Accuracy
3.1.1 LSTM

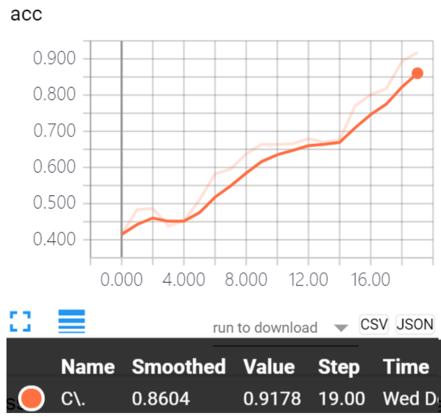

fig 3.1.1 Training Accuracy

fig 3.1.2 Testing Accuracy

fig 3.1.3 Testing Accuracy

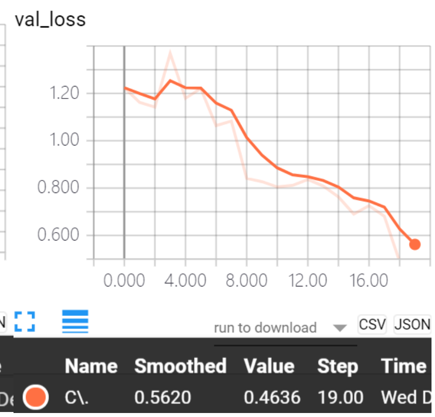

fig 3.1.4 Testing Loss

This result can be the best result in this project. There is no overfitting happened.

3.1.2 Bi- directional LSTM

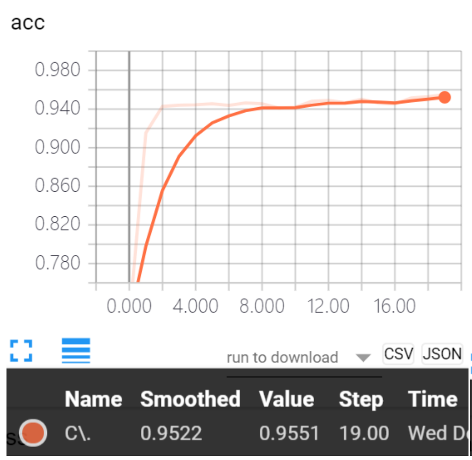

fig 3.2.1 Training Accuracy

fig 3.2.2 Testing Accuracy

fig 3.2.3 Testing Accuracy

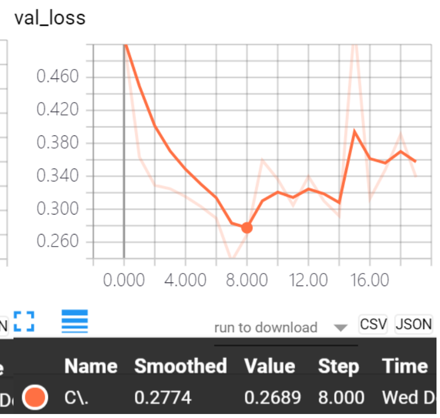

fig 3.2.4 Testing Loss

We should stop our iteration when epochs=8 because of the overfitting will happen after it. We can see the training accuracy is higher than testing accuracy and training loss is higher than testing loss. What's more, when epochs=15, the testing loss begin to arise.

3.1.2 CNN+LSTM

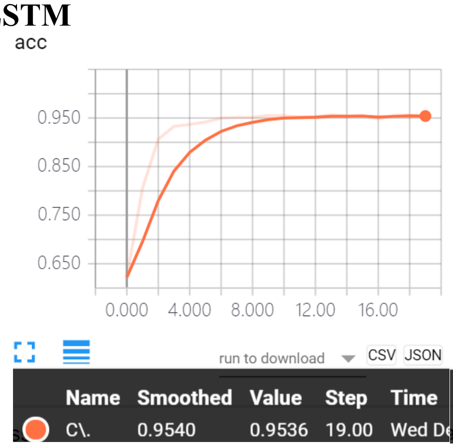

fig 3.3.1 Training Accuracy

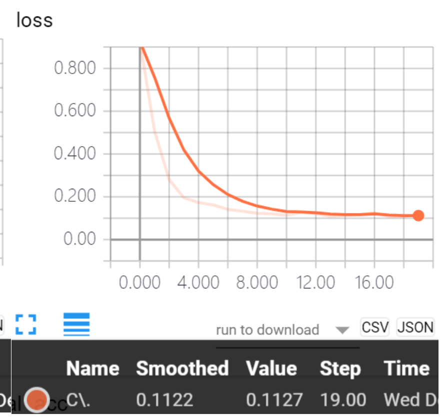

fig 3.3.2 Testing Accuracy

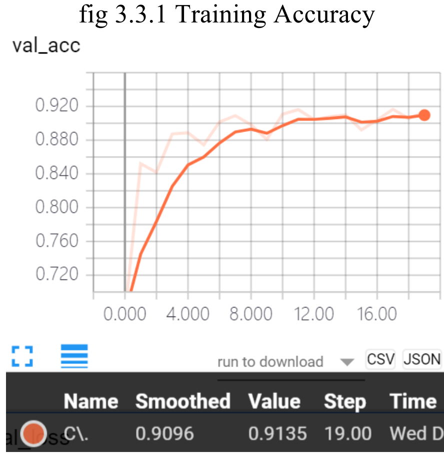

fig 3.3.3 Testing Accuracy

fig 3.3.4 Testing Loss

3.1.2 CNN+Bi- directional LSTM

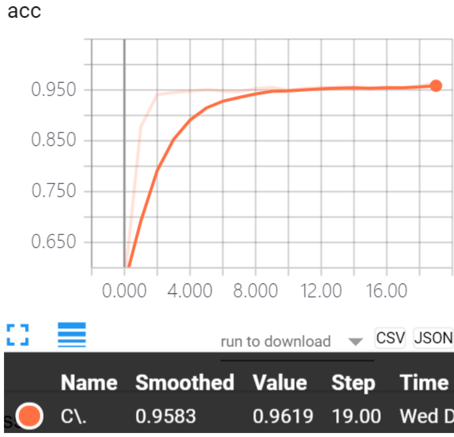

fig 3.4.1 Training Accuracy

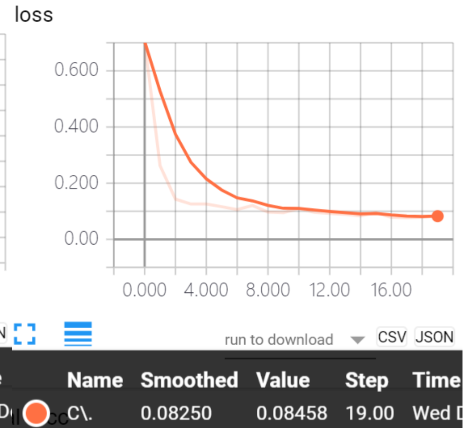

fig 3.4.2 Testing Accuracy

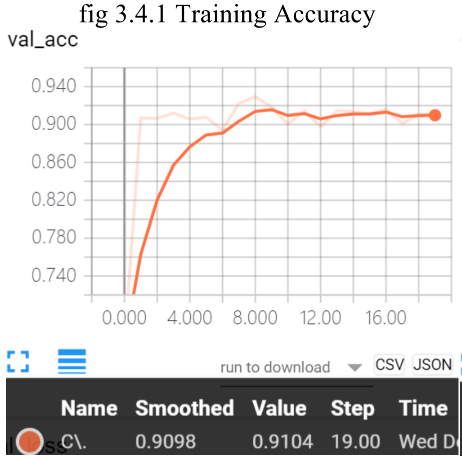

fig 3.4.3 Testing Accuracy

fig 3.4.4 Testing Loss

After analyzing those four groups of results, we may find that there is some overfitting may happen but LSTM and bi-directional LSTM can be seen as an acceptable result. Compare to them, CNN+LSTM and bi-directional LSTM have a significant overfitting. Therefore, those models are more difficult to be used in this dataset. Actually, we cannot test those model by this dataset.

3.2 Confusion Matrix

Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).

Pred	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
True						
LAYING	510	0	0	0	0	27
SITTING	0	363	124	4	0	0
STANDING	0	54	469	7	0	2
WALKING	0	0	0	435	61	0
WALKING_DOWNSTAIRS	0	0	0	4	416	0
WALKING_UPSTAIRS	0	0	1	9	51	410

fig 3.5.1 Confusion Matrix of LSTM

Pred / True	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	537	0	0	0	0	0
SITTING	5	386	96	0	0	4
STANDING	0	76	455	1	0	0
WALKING	0	2	1	469	14	10
WALKING_DOWNSTAIRS	0	1	0	3	398	18
WALKING_UPSTAIRS	0	0	0	10	7	454

fig 3.5.2 Confusion Matrix of Bi-directional LSTM

Pred / True	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	519	0	0	0	0	18
SITTING	1	430	57	0	0	3
STANDING	0	122	410	0	0	0
WALKING	0	0	0	460	31	5
WALKING_DOWNSTAIRS	0	0	0	0	410	10
WALKING_UPSTAIRS	0	0	0	0	8	463

fig 3.5.3 Confusion Matrix of CNN+LSTM

Pred / True	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	510	0	0	0	0	27
SITTING	0	395	77	0	0	19
STANDING	0	95	437	0	0	0
WALKING	0	0	0	491	3	2
WALKING_DOWNSTAIRS	0	0	0	1	412	7
WALKING_UPSTAIRS	0	0	0	8	25	438

fig 3.5.4 Confusion Matrix of CNN+Bi-directional LSTM

By analyzing those four confusion matrixs, we can found that:

(1) Comparing LSTM to CNN+LSTM, LAYING, SITTING, WALKING, WALKING_DOWNSTAIRS shows an improvement of prediction accuracy while STANDING and WALKING_UPSTAIRS shows an worse prediction accuracy.

(2) There are same items of improvement when comparing LSTM to Bi-directional LSTM.

(3) There is a bad result of CNN+Bi-directional LSTM. Most of prediction accuracy are decrease than before.

3.3 Data Visualization

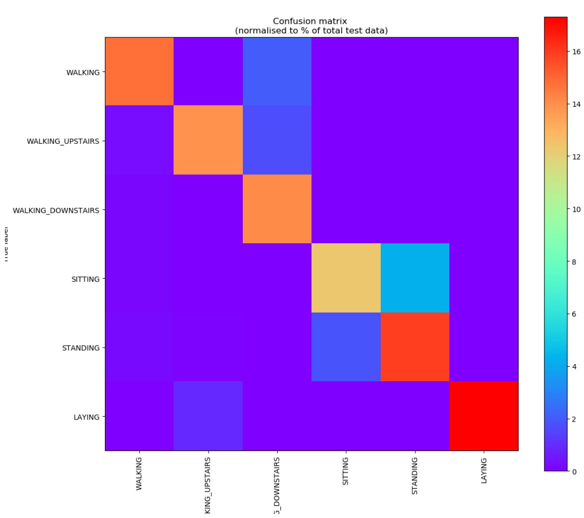

fig 3.6.1 LSTM

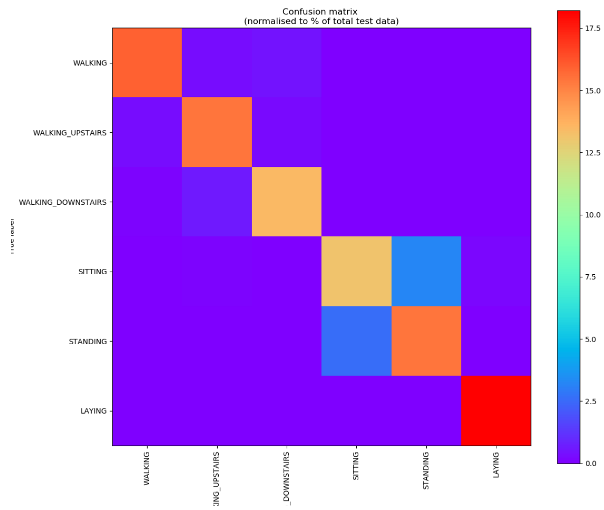

fig 3.6.2 Bi-directional LSTM

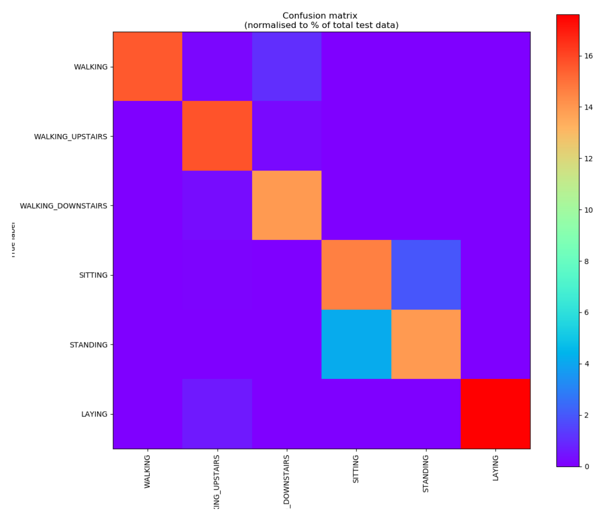

fig 3.6.3 CNN+LSTM

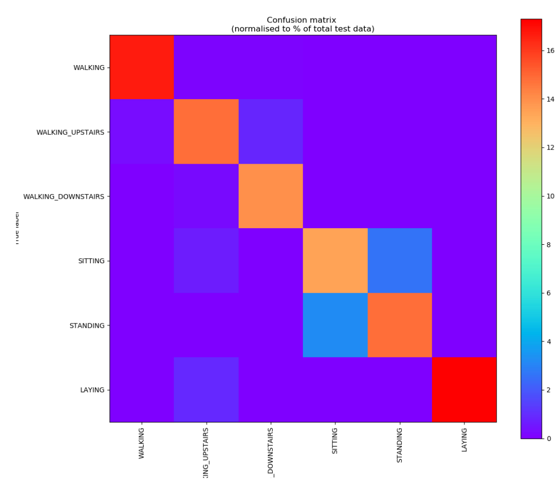

fig 3.6.4 CNN+Bi-directional LSTM

4. Dataset
4.1 introduction
The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.

The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain.

For each record in the dataset it is provided:
- Triaxial acceleration from the accelerometer (total acceleration) and the estimated body acceleration.
- Triaxial Angular velocity from the gyroscope.
- A 561-feature vector with time and frequency domain variables.
- Its activity label.
- An identifier of the subject who carried out the experiment.

5. Improvement
5.1 Dataset
According to the result, there are some serious overfitting will happen when model become more complex than before. (Compare LSTM to CNN+LSTM). The data are 1 dimension, therefore it is difficult to do data augmentation. Early stop and decreasing the complexity of model are not a good choice for us. Therefore, Next time I will try to change a dataset.

I found that there is another larger dataset which is called WISDM: Wireless Sensor Data mining. This dataset contains data collected through controlled, laboratory conditions. If you are interested in "real world" data, please consider our Actitracker Dataset.

Number of examples: 1,098,207
Number of attributes: 6
Missing attribute values: None
Class Distribution
Walking: 424,400 (38.6%)
Jogging: 342,177 (31.2%)
Upstairs: 122,869 (11.2%)
Downstairs: 100,427 (9.1%)

Sitting: 59,939 (5.5%)
Standing: 48,395 (4.4%)

5.2 Model

I found that there is some improvement of accuracy between LSTM and CNN+LSTM. But after analyzing the result of bi-directional LSTM and CNN+ bi-directional LTSM, there is no such improvement of accuracy. In the future, I want to know why there is no significant improvement when the neural networks are similar.

The first thing I want to do is change my dataset to eliminate the influence from it. I want to know if there is a larger dataset and no overfitting, will CNN+ bi-directional LSTM can perform better than bi-directional LSTM? Or bi-directional LSTM is just suit for working itself?

What's more, I find that my CNN is not complete. I see another paper about how to apply CNN to human activity recognition. I will try this model in the future to see if it has some influence for accuracy.

6. Learning from the final project

Actually, I think that I still have a long distance to other students. The first shortcoming of my final project is that My model is easier than them. I just use an easy network structure and have no paper as a base. In the future, I will try to realize the structure in paper. I should understand others paper and model before I try to create myself model. If I do not do this, my model is just like float in the sky which with any base. The second shortcoming is that I still have some confusion of neural network, I think I should learn deep learning in a systematic way and reviewing the knowledge which we learnt in our class.