Human Activity Recognition Based on Recurrent Neural Networks

A Preprint

John Pravin Arockiasamy
Department of Information Technology
The University of Stuttgart
Stuttgart, 70569
st180270@stud.uni-stuttgart.de

Ayush Mittal
Department of Electrical Engineering
The University of Stuttgart
Stuttgart, 70569
st176425@stud.uni-stuttgart.de

February 12, 2023

Abstract

Human Activity recognition is a branch of research that predicts human activities based on the time series dataset like the Human Activities and Postural Transitions (HAPT) dataset. In this deep learning paper, we have highlighted the input pipeline to process, and the models to predict the 6 activities and their corresponding transitions of the HAPT time series data. Also, we have constructed different metrics to evaluate the model's performance. In the end, the proposed models have an accuracy of around 93 in percent.

1 Introduction

Human activity recognition is the way to predict human motions like walking, sitting, standing, lying, etc which can be used further to assist humans in multiple sectors and to improve human-computer interactions. In the medical sector, successful activity recognition would open up a wide range of applications such as understanding old people's mental and physical illness based on the changes in their daily activities. Day-to-day individual human activities data can be captured and analyzed using an external sensor or body-worn sensor data. In addition to recording the activity data, it is also useful to record the transitions between them. Recording individual activities and their transitions can be a hectic and challenging problem involving a rightful sensor choice. Based on the multiple research study, it is recommended to use the body-worn sensor instead of the external sensor to capture the activity data as the external sensor is prompted by noise and range limitations. In this paper, we have discussed a generic pipeline from pre-processing to classifying the activities and their transition of data.

2 Dataset

For human activity recognition, we have taken the Human Activities and Postural Transitions (HAPT) dataset. The dataset was captured with 30 users performing three static activities (standing, sitting, lying) and 3 dynamic activities (walking, walking downstairs, and walking upstairs) and their transitions on a Samsung phone mounted on the waist. The captured sensor data consists of a 3-axis accelerometer and 3-axis gyroscope data which is then pre-processed to remove noise. The sensor data is sampled to have a constant frequency of 50Hz. The labeling of this time-series data was later done manually.

CLASS	DESCRIPTION
0	WALKING
1	WALKING_UPSTAIRS
2	WALKING_DOWNSTAIRS
3	SITTING
4	STANDING
5	LAYING
6	STAND_TO_SIT
7	SIT_TO_STAND
8	SIT_TO_LIE
9	LIE_TO_SIT
10	STAND_TO_LIE
11	LIE_TO_STAND

Figure 1: Description of classes in HAPT dataset

3 Preprocessing

3.1 Unlabelled data:

In the HAPT dataset, it is observed that there are a few data points that are not labeled. In order to avoid the classification error due to unlabelled data, we have constructed the input pipeline to retrieve only the labeled data ignoring the unlabelled data.

3.2 Input Normalization:

In the HAPT dataset, it is observed that there are offsets and variances among the user's data due to different mounting characteristics of the phone on the waist, and removing such offsets and variances could provide better performance. Thus, the Z-score normalization is used in the input pipeline to normalize 6-channel input time-series data with zero mean and unit variance.

3.3 Dataset Split:

In the input pipeline, we have performed a 70%/10%/20% of the train/validation/test split of the input time-series data. Such split is achieved by retrieving from the user-01 to user-21 data as train, from the user-28 to user-30 data as validation, and from the user-22 to user-27 data as a test.

3.4 Sliding window:

The HAPT input (3-axis accelerometer and 3-axis gyroscope) data cannot be fed into the model as it learns the correlation between the consecutive measurements not the sequence of raw data. Thus, the input pipeline uses the sliding window to cut the dataset into windows of fixed-length (250 timestamps) input sequences which is used as a data augmentation as well.

3.5 TF-Records:

In order to process the HAPT dataset efficiently, we have serialized the 6-channel train, test, and validation data and their corresponding labels and stored them as TFRecord data.

4 Model

A Recurrent Neural Network is a type of artificial neural network that can be used to model sequence data. RNNs are used for time-series data due to the ability to remember the past as the output from the previous step is fed in as an input. The major drawback of vanilla RNN models is the lack of ability to train when long-term temporal dependencies are required. Thus, for this HAPT dataset, we have selected long short-term memory (LSTM) and gated recurrent unit (GRU) models as shown in Figure 2.

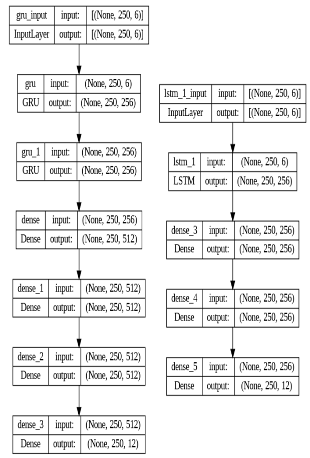

Figure 2: Architecture of small GRU (Left) and small LSTM (Right)

4.1 Long Short-Term Memory (LSTM):

LSTM is the type of RNNs that is capable of training on long-term temporal dependencies. Unlike RNNs, the LSTM controls the flow of information in and out of a cell using three gates (the input gate, the output gate, and the forget gate) and the cell remembers the values over arbitrary time intervals. Thus, making the LSTM work with the prediction of time series with uncertain duration.

4.2 Gated Recurrent Unit (GRU):

The Gated Recurrent Unit is also a special variant of recurrent neural network architecture. The GRU model has a similar structure and working mechanism as compared to the LSTM model but with only 2 gates (reset and update gates).

5 Results and Conclusion

5.1 Training and Evaluation:

Loss: The sparse cross-entropy loss is selected as the loss function with the logit variable to be True. The formulae for the sparse cross-entropy can be found below.

$$CE = -\sum_{i}^{C} t_i \log(c_i)$$

where t_i is the label and c_i is the logit output of specific class i.
Metrics: The sparse cross-entropy accuracy and f1_score are selected as the metrics to evaluate the performance of the model on training, test, and validation data. The confusion matrix of 12*12 dimensions is selected to provide the classification details on the test data.

Optimizer: Adam optimizer is selected as the optimizer for the modal to train as the Adam optimizer combines the benefits of RMSProp and AdaGrad.

5.2 Observations:

The results on test dataset are shown in Figure 3. orginial and predicted labels for a sequence of 8,000 timestamps is visualized and the corresponding confusion matrix is shown in Figure 4. Our best model which is GRU provided an accuracy of **93.27%** and due to the unbalanced amount of data in different categories, we could observe that the model preforms significantly worse in the categories with less data.

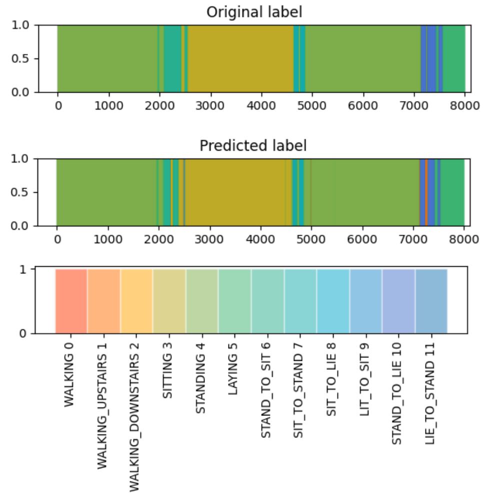

Figure 3: Visualization of ground truth and predicted labels

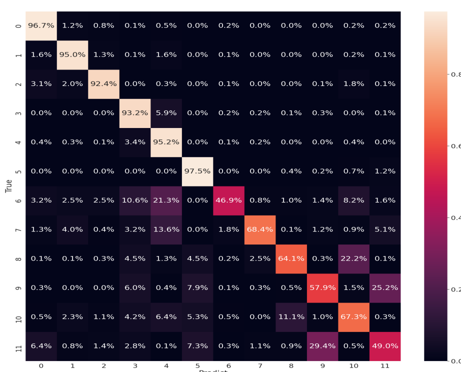

Figure 4: Confusion Matrix of LSTM model

Also, we could observe that the model has ability to accurately predict the 6 basic activities but poor at predicting the postural transitions like sit-to-lie or lie-to-sit due to the lack of samples.

References

[1] Salem, Fathi M. "Gated RNN: The Gated Recurrent Unit (GRU) RNN." Recurrent Neural Networks, 2021, 85–100. https://doi.org/10.1007/978-3-030-89929-5_5.

[2] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long Short-Term Memory." Neural Computation 9, no. 8 (1997): 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

[3] Yeung, Kei-wai. "Sequence Processing with Recurrent Neural Networks," n.d. https://doi.org/10.14711/thesis-b447453.

[4] Nweke, Henry Friday, Ying Wah Teh, Mohammed Ali Al-garadi, and Uzoma Rita Alo. "Deep Learning Algorithms for Human Activity Recognition Using Mobile and Wearable Sensor Networks: State of the Art and Research Challenges." Expert Systems with Applications 105 (2018): 233–61.

[5] Chen, Kaixuan, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. "Deep Learning for Sensor-Based Human Activity Recognition." ACM Computing Surveys 54, no. 4 (2021): 1–40. https://doi.org/10.1145/3447744.