Human Activity Recognition

A Preprint

Peng Gu
Electromobility
University of Stuttgart
70569, Stuttgart
st169623@stud.uni-stuttgart.de

Shuai Zhang
Electromobility
University of Stuttgart
70569, Stuttgart
st169622@stud.uni-stuttgart.de

February 3, 2022

Abstract

Over the past decades, human activity recognition(HAR) has grabbed considerable research attentions due to its prominent applications such as smart home health care. In this paper, we propose two effective models to accurately identify the status of individuals' daily activities.

1 Introduction (Shuai ZHANG)

HAR is an important area of research in ubiquitous computing, human behaviour analysis, and human-computer interaction. It is important to capture and record the daily activities of individuals for many reasons, for example, to better understand the health status of patients [1].Sensor-based systems are one of the main streams of HAR systems. Due to the popularity of smart devices and the Internet of Things, sensors can be integrated into wearable devices such as cell phones to record information about human activities in an uninterrupted and non-invasive manner [2] [3].

2 Input Pipeline (Shuai ZHANG)

2.1 HAPT Dataset

The data of the dataset came from an experiment conducted by 30 volunteers aged 19-48. Each person performed six activities (walking, walking-upstairs, walking-downstairs, sitting, standing, lying). There are also 6 postural transitions between 3 static activities, namely sitting, standing, and lying. Subsequently, 3-axial linear acceleration and 3-axial angular velocity were recorded at a constant rate of 50 Hz using embedded sensors of a smartphone worn around the waist. In this work, we further divided the dataset into training, validation, and test datasets with a ratio of $7:2:1$.

2.2 Preprocessing

First, we removed all unlabeled data from the dataset. Second, for each channel of the remaining data, we independently performed z-score normalization to remove the biases and different variances of the data. Next, we used the sliding window technique. On the one hand, we ensured that the data contained sequences of equal length after using the sliding window. On the other hand, this can also be considered as a data augmentation technique. We set two variable parameters, the window size and the window non-overlapping ratio, which were set to 250 and 0.5 by default respectively.

2.3 Generating TFRecord Files

To load the data more efficiently, we serialized the data of training, validation, and test sets and created the corresponding TFRecord files.

3 Models (Peng GU)

A Recurrent Neural Network (RNN) is a type of neural network that is well-suited to time series data. RNN processes a time series step-by-step, while maintaining the internal state from previous time step to next time step. Therefore, for this HAR task, RNN was selected as the basic network to solve the time series problem. Furthermore, two RNN architectures, namely long short-term memory (LSTM) and gated recurrent unit (GRU), were implemented for training.

3.1 LSTM Model

Traditional recurrent backpropagation is very inefficient because of storing information over extended time intervals. To solve this problem, LSTM learns to truncate the gradient by using multiplicative gate units. These gate units learn to open and close access to the constant error flow. As a result, LSTM is much more efficient. It has great advantage on computational complexity [4].

3.1.1 Model Architecture

The model is mainly combined with three parts. The first part is LSTM layers with drop out, which is a regularization method to prevent overfitting. The second part is dense layers with rectified linear unit (ReLU) as activation function. The last part is the prediction layer with softmax as activation function.

3.1.2 Objective

The problem is treated as sequence-to-sequence classification task. The LSTM model returns sequences as output. The loss function is SparseCategoricalCrossentropy between the prediction labels and the true labels. The optimizer is Adam with learning rate equals 0.001.

3.2 GRU Model

Similar to LSTM, GRU is also a gating-based mechanism of RNN. Therefore, GRU model is also very efficient and has advantage on computational complexity. What's more, GRU has fewer parameters than LSTM. In comparison with LSTM, GRU reduces the gating signals to two, namely update gate and reset gate [5].

3.2.1 Model Architecture

The architecture is similar to that of the LSTM models. The first part is a stack of GRU layers with drop out to prevent overfitting. The second part is a stack of dense layers with rectified linear unit (ReLU) as activation function. The last part is the prediction layer. For classification task, the softmax activation function is selected and the output units equals the number of different classes.

3.2.2 Objective

The GRU model also solve the sequence-to-sequence classification task and returns sequences as output. The loss function is SparseCategoricalCrossentropy between the prediction labels and the true labels. The optimizer is Adam with learning rate equals 0.001.

4 Hyperparameter Optimization (Peng GU)

The hyperparameters, such as window size, shift window size for input data, number of layers, number of units and drop out rate for models, were tuned to obtain a better performance.
For LSTM model, the hyperparameter optimization result is shown in table 1.

For GRU model, the hyperparameter optimization result is shown in table 2.

Table 1: Hyperparameters of LSTM model

Name	Description	Value
num_rnn_layers	Number of LSTM layers	1
num_rnn_neurons	Number of units in each LSTM layer	256
dropout_rate	Drop out rate in each LSTM layer	0.014457
num_dense_layers	Number of dense layers	2
num_dense_neurons	Number of units in each dense layer	256

Table 2: Hyperparameters of GRU model

Name	Description	Value
num_rnn_layers	Number of GRU layers	2
num_rnn_neurons	Number of units in each GRU layer	256
dropout_rate	Drop out rate in each GRU layer	0.14
num_dense_layers	Number of dense layers	3
num_dense_neurons	Number of units in each dense layer	512

5 Results (Peng GU)

5.1 LSTM Result

The LSTM model reached accuracy of 92.4%. The confusion matrix and normalized confusion matrix on the test dataset are shown in figure 1 and figure 2.

5.2 GRU Result

The GRU model reached accuracy of 93.7%. The confusion matrix is shown in figure 3 and the normalized confusion matrix is shown in figure 4.

6 Conclusion (Peng GU)

This study aimed to accomplish sequence-to-sequence classification task on the time series dataset. The raw data were processed to fixed samples with the same length. Moreover, Z-score method was applied on the dataset to solve the bias problem. Two RNN models, namely LSTM model and GRU model, were implemented to train on the dataset. Hyperparameters were optimized by tuning the model. As a result, the LSTM model reached accuracy of 92.4% and the GRU model reached accuracy of 93.7%. However, the performance on the postural transition needs to be improved in further work.

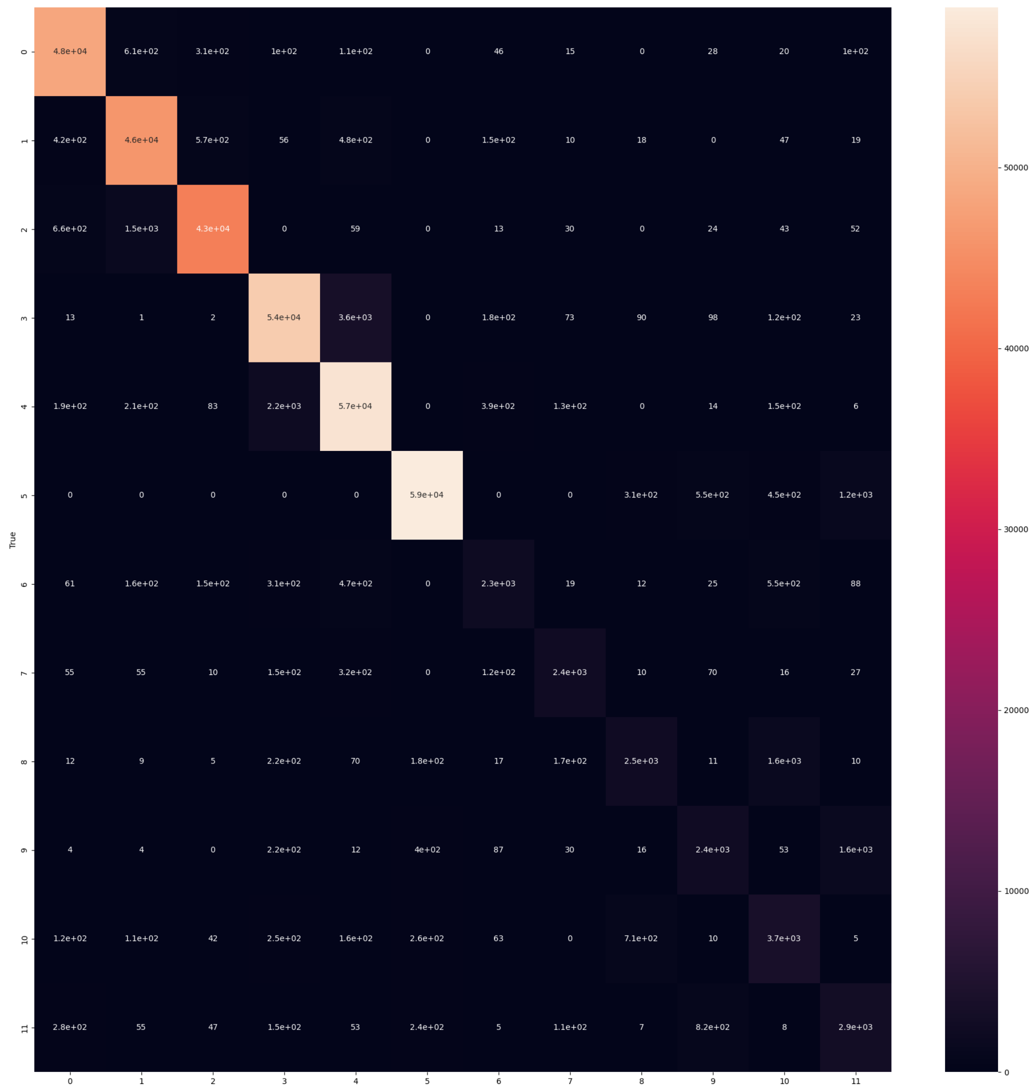

Figure 1: Confusion matrix, test by LSTM model

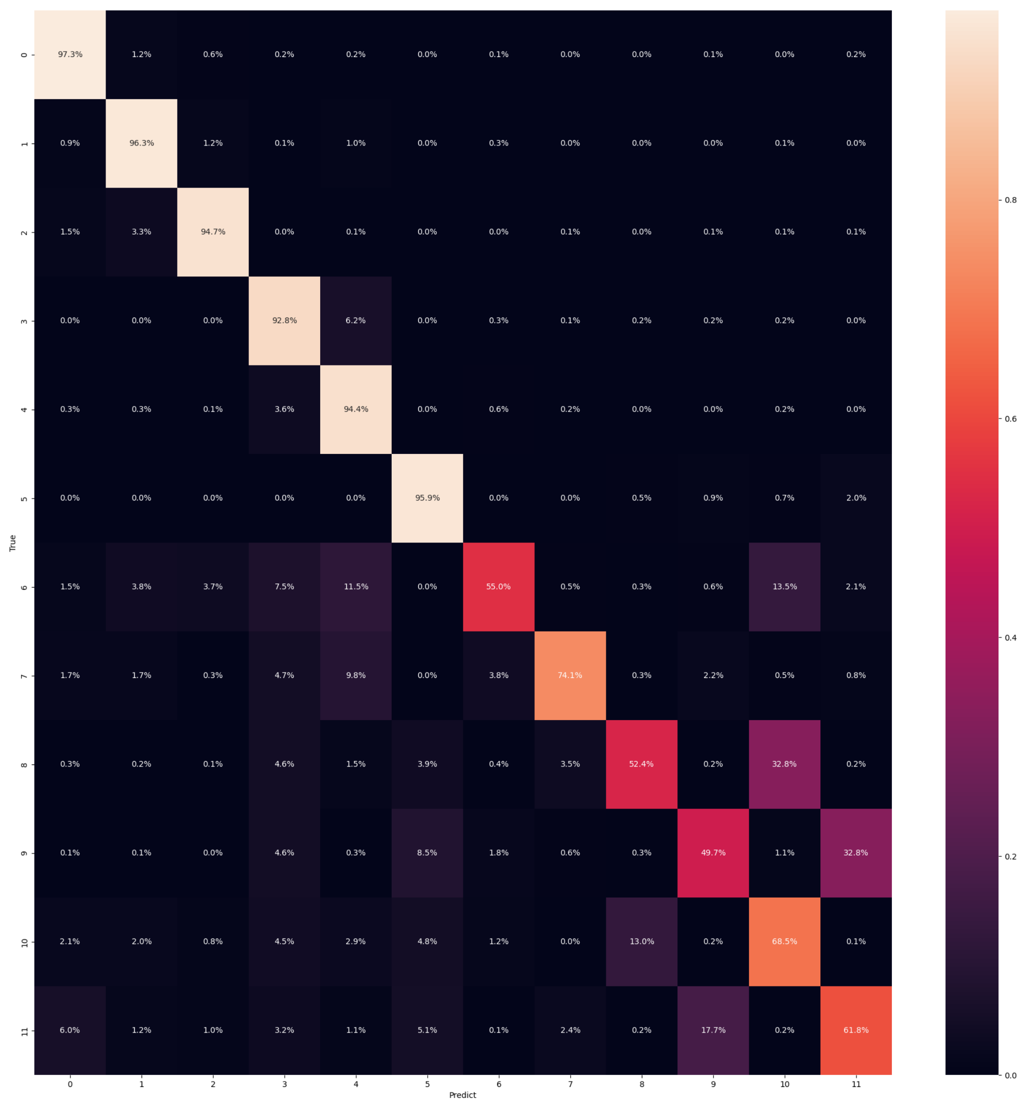

Figure 2: Normalized confusion matrix, test by LSTM model

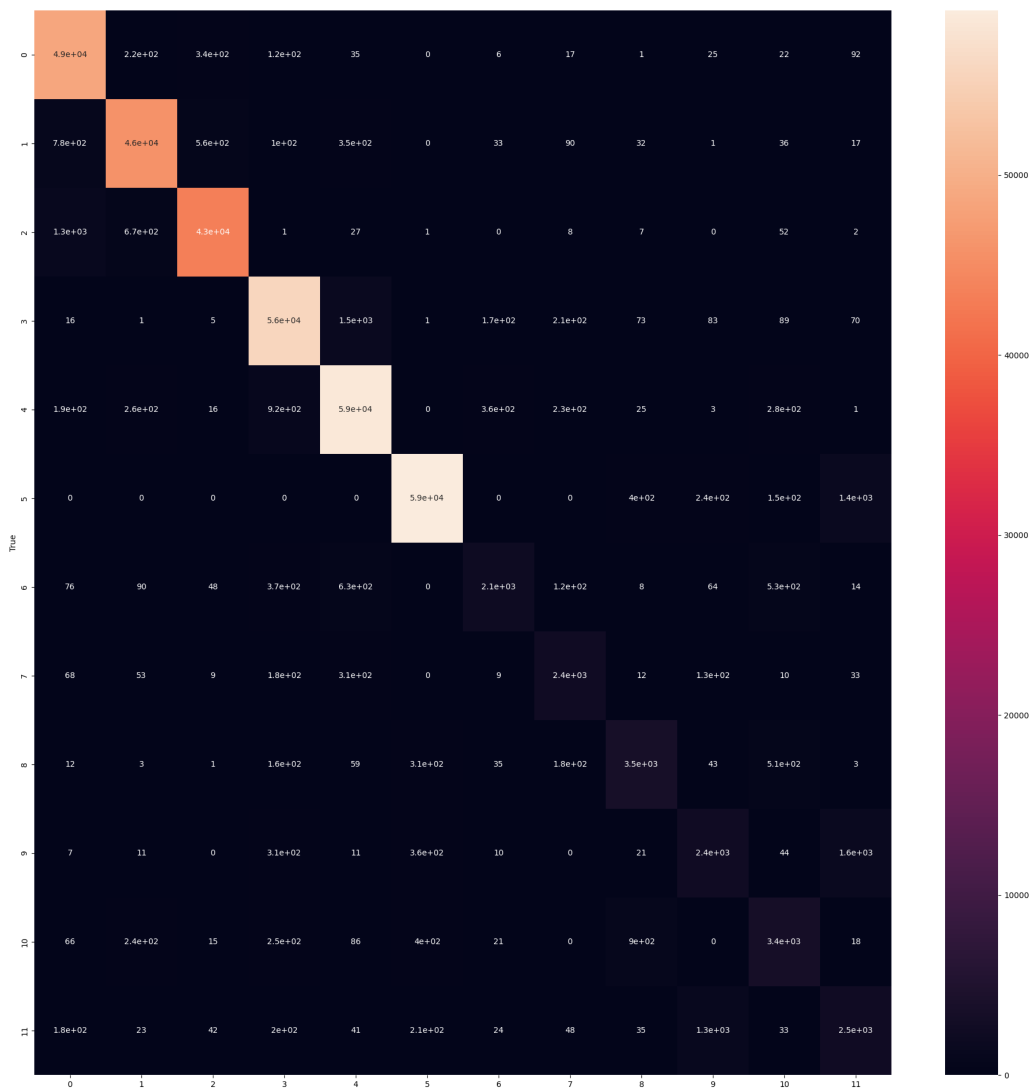

Figure 3: Confusion matrix, test by GRU model

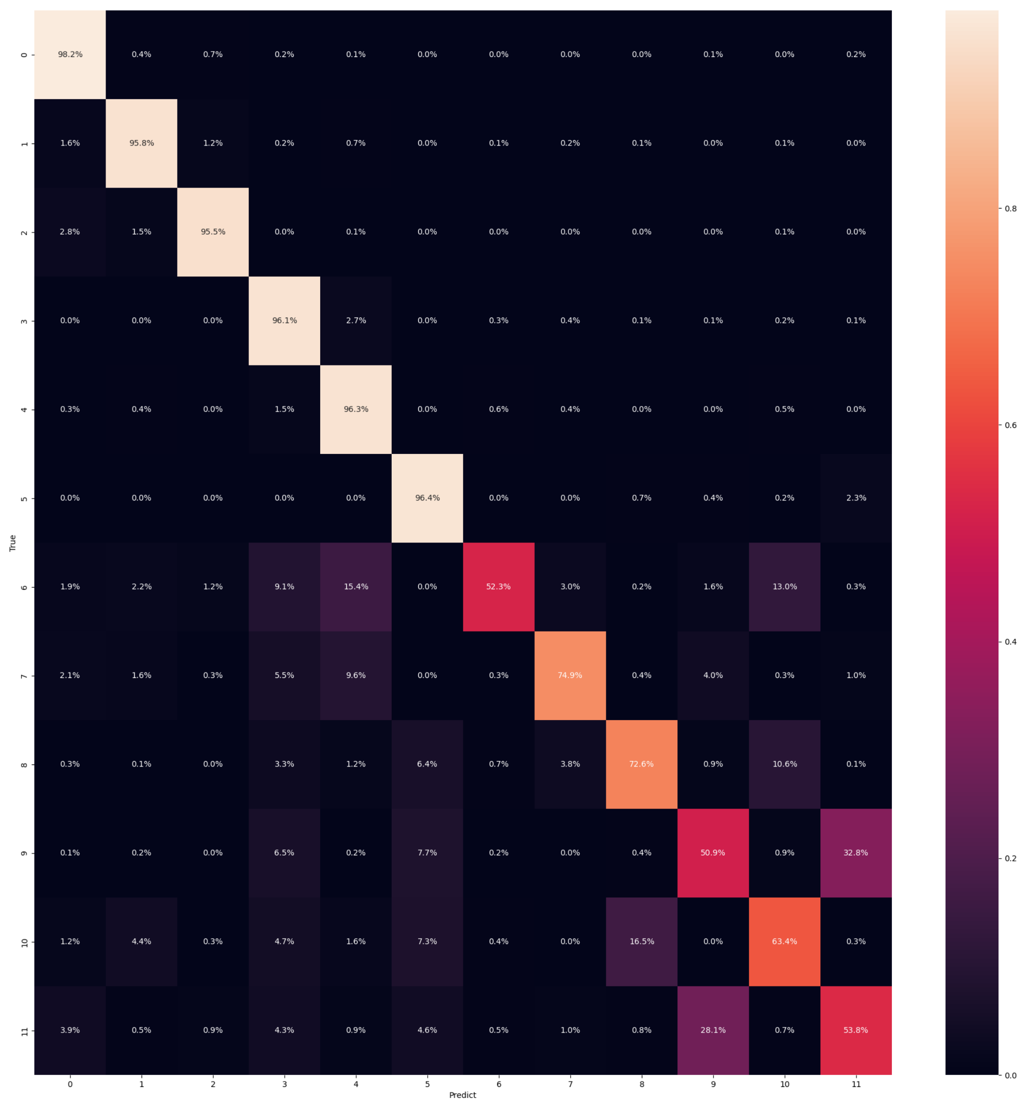

Figure 4: Normalized confusion matrix, test by GRU model

References

[1] Daniele Ravi, Charence Wong, Benny Lo, and Guang-Zhong Yang. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In *2016 IEEE 13th international conference on wearable and implantable body sensor networks (BSN)*, pages 71–76. IEEE, 2016.

[2] Henry Friday Nweke, 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*, 105:233–261, 2018.

[3] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Computing Surveys (CSUR)*, 54(4):1–40, 2021.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[5] Rahul Dey and Fathi M. Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1597–1600, 2017.