LITERATURE REVIEW

Long Short-Term Memory (LSTM)

AREA

Deep Learning

Submitted to

Prof. Sonia Abraham

Submitted by

Anlin Albert

Literature Review – Seminar

1. Das, S., Partha, S. B., & Imtiaz Hasan, K. N. (2020). Sentence Generation using LSTM Based Deep Learning. 2020 IEEE Region 10 Symposium (TENSYMP).

Abstract: The process of predicting pertinent words in a particular order is served by sentence generation. This study aims to develop a process for producing sentences while upholding correct grammatical structure.

Using the Long Short-Term Memory (LSTM) architecture creates a phrase creation system here. The fundamentals of word embedding are generally followed by the system, where words from the dataset are tokenized and transformed into vector shapes.

Following processing, a layer of long short-term memory is used to store these vectors. After each repetition, the system generates a new set of words. As a result of this process, a sentence or passage will eventually be formed using pertinent words. In comparison to other existing approaches, the system's results are fairly compelling.

Methodology: A suitable dataset with a large number of words is used to train the system. A vocabulary is created using special terms. To create new words that are appropriate for the situation, the labels and features are retrieved, and long short-term memory architecture is then utilized. The general steps are broken down into a few key modules, which are discussed in more detail below.

* Data collection and description

Collect data and prepare model

* Processing data for model

Process collected data and train the model

* + Creating vocabulary

Here the words are split and a vocabulary of unique words is created

* + Tokenization

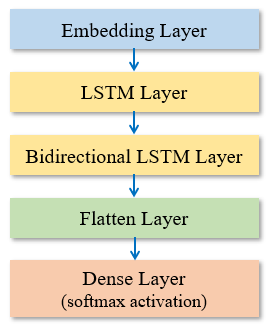
Each word gets tokenized meaning each of the words can be uniquely identified using a number

* + N-grams generation
  + Padding

N-grams generated may be of different lengths, here we use padding to acquire uniformity in length

* + Retrieving labels and features
  + One-hot encoding

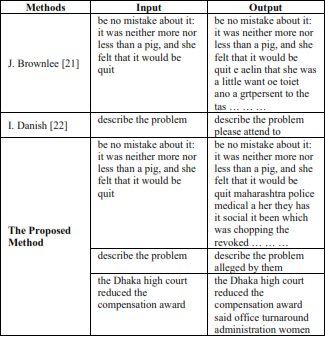
Proposed Model: In this paper, the proposed model has several sequential layers



* Embedding layer: In this layer, a set of words is mapped into vector forms to improve the ability of neural networks because working on numerical data is much easier
* LSTM
* Bidirectional LSTM: Bidirectional LSTM has two hidden states working in opposite directions and hence can work on both past and future states at a time
* Flatten layer: This layer takes the output of the previous layer and puts the value in a single vector
* Dense layer: The dense layer in the model uses a softmax activation function which is nonlinear and uses adam optimizer to fit features and labels
* Generating new word: After training the model, input is given to it in tokenized form. The model predicts a new word after each iteration maintaining the context. After generating every new word, the word also is added to the previous input and the new combination is considered the next input.

Result: 80 percent of the dataset is part of the system's training set. The corresponding loss and accuracy rate is noticed after the training data has been fitted into the model. From Fig, it is clear that the rate of loss is steadily declining while the rate of accuracy per epoch is steadily rising.

The table below shows that the first compared model generates words that aren't even real. Additionally, the meaning of the model's output is not particularly clear. The suggested model produces more insightful results for the same input sequence. No non-existent words are generated by the model either. They compared the model in the second comparison and generates an absurd sequence. To make the resilience of the system more understandable, it also shows a third example produced by the suggested system.



Conclusion: This study presents the idea of developing a deep learning-based model that can produce novel sentences. Word embedding and the Long Short-Term Memory (LSTM) architecture, a modified version of recurrent neural networks, serve as the foundation for all of the methodologies (RNN). The text data are transformed into vector form using word embedding since the neural network finds it easier to operate with numerical data. LSTM and Bidirectional LSTM networks are used to preserve the correct context. Because it can work on both the present and the past at once, the bidirectional LSTM is a crucial layer for the model. The model was effectively trained, and the outcomes were carefully examined and contrasted with those from other models.

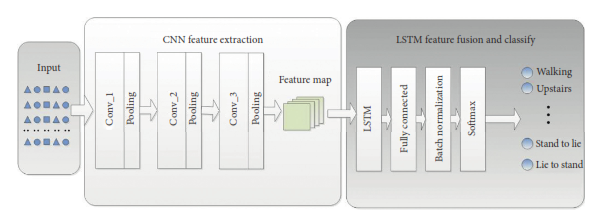
Future Work: In subsequent work, the model can be improved even more so that it can be applied to creating dynamic chatbot suggestions or summarising material.

1. Wang, H., Zhao, J., Li, J., Tian, L., Tu, P., Cao, T., … Li, S. (2020). Wearable Sensor-Based Human Activity Recognition Using Hybrid Deep Learning Techniques. Security and Communication Networks, 2020, 1–12.

Introduction: Human behavior recognition (HAR) is the detection, interpretation, and recognition of human behaviors, which can use smart health care to actively assist users according to their needs. Human behavior recognition has wide application prospects, such as monitoring in smart homes, sports, game controls, health care, elderly patients care, bad habits detection, and identification. It plays a significant role in in-depth study and can make our daily life smarter, safer, and more convenient. This work proposes a deep learning-based scheme that can recognize both specific activities and the transitions between two different activities of short duration and low frequency for health care applications.

Dataset: This paper adopts the international standard Data Set, Smartphone-Based Recognition of Human Activities, and Postural Transitions Data Set to conduct an experiment, which is abbreviated as HAPT Data Set. The data set is an updated version of the UCI Human Activity Recognition Using popularity Data set. It provides raw data from smartphone sensors rather than preprocessed data and collects data from accelerometer and gyroscope sensors. In addition, the action category has been expanded to include transition actions. The HAPT data set contains twelve types of actions. A total of 815,614 valid pieces of data are used here.

Proposed method:



The above figure displays the three-part architecture design for the strategy suggested in this paper. The initial step is the preprocessing and transformation of the raw data, which combines the raw acceleration and gyroscope data into a two-dimensional array that resembles an image. The composite image must then be entered into a CNN network with three layers so that it can automatically detect from the activity image, extract the motion features, abstract the features, and then map them into the feature map. The third step involves feeding the feature vector into the LSTM model, establishing a connection between time and action sequence, and then introducing the entire connection layer to fuse the numerous features. Additionally, Batch Normalization (BN) is presented. BN can normalize the data in every layer before sending it to the Softmax layer for action classification.

This paper also examines transition actions in addition to typical fundamental acts. Transition actions are present in a few accessible data sets. For this reason, the HAPT Data Set, also known as the Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set, is used in this paper to experiment. A newer version of the UCI Human Activity Recognition Using Popularity Data collection is used in this dataset. Instead of preprocessed data, it offers raw data from smartphone sensors. Additionally, transition actions have been added to the category of activities. All of the data without labels were removed after the initial processing of the original data. In the end, 815,614 valid data points were gathered. There is a sizable disparity in data volume between transition action and basic action as a result of the low frequency and brief duration of transition action as well as the high frequency and lengthy duration of fundamental action. The six transition actions account for just about 8% of the overall data, which is significantly less than the data amount of the other basic acts. The initial collection of data is divided into three components: a training set, a verification set, and a test set. The training set is used to train the model, the verification set to alter its parameters, and the test set to assess the quality of the resultant model.

1. Agarwal, P., & Alam, M. (2020). A Lightweight Deep Learning Model for Human Activity Recognition on Edge Devices. Procedia Computer Science, 167, 2364–2373.

Introduction: Here the architecture for the proposed Lightweight model is developed using Shallow Recurrent Neural Network (RNN) combined with Long Short Term Memory (LSTM) deep learning algorithm. then the model is trained and tested for six HAR activities on resource-constrained edge devices like RaspberryPi3, using optimized parameters. The experiment is conducted to evaluate the efficiency of the proposed model on the WISDM dataset containing sensor data of 29 participants performing six daily activities: Jogging, Walking, Standing, Sitting, Upstairs, and Downstairs. And lastly, the performance of the model is measured in terms of accuracy, precision, recall, f-measure, and confusion matrix and is compared with certain previously developed models.

Dataset: Here Android smartphone having an inbuilt accelerometer is used to capture tri-axial data. The dataset consists of six activities performed by 29 subjects. These activities include walking, upstairs, downstairs, jogging, standing, and sitting. Each subject performed different activities by carrying a cell phone in the front leg pocket. A constant Sampling rate of 20 Hz was set for the accelerometer sensor. A detailed description of the dataset is given in table 1 below.

Total no of samples: 1,098,207

Total no of subjects: 29

Activity Samples: Percentage

Walking 4,24,400 38.6%

Jogging 3,42,177 31.2%

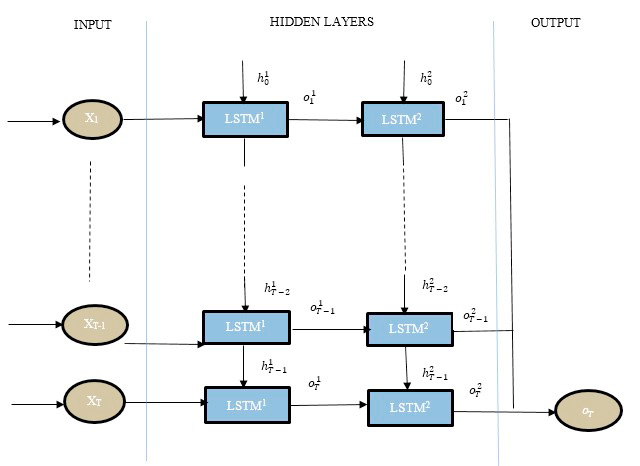
Upstairs 1,22,869 11.2%

Downstairs 1,00,427 9.1%

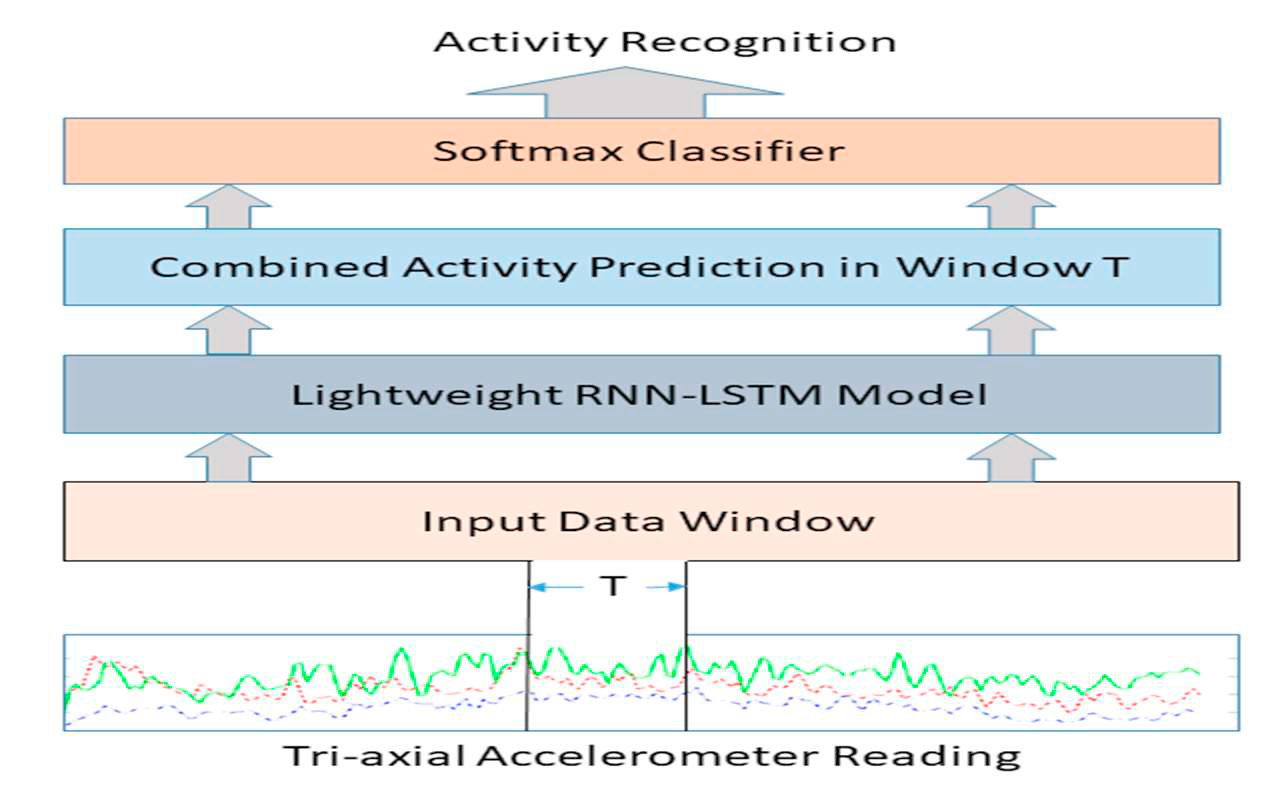
Sitting 59,939 5.5%

Standing 48,397 4.4%

Proposed method: RNN and LSTM are used to create the proposed model. With only two hidden layers and 30 neurons, it has a shallow structure that makes it practical to install on edge computing devices such as IoT boards (Raspberry Pi, Audrino, etc.), Android, and iOS-based resource-constrained devices.



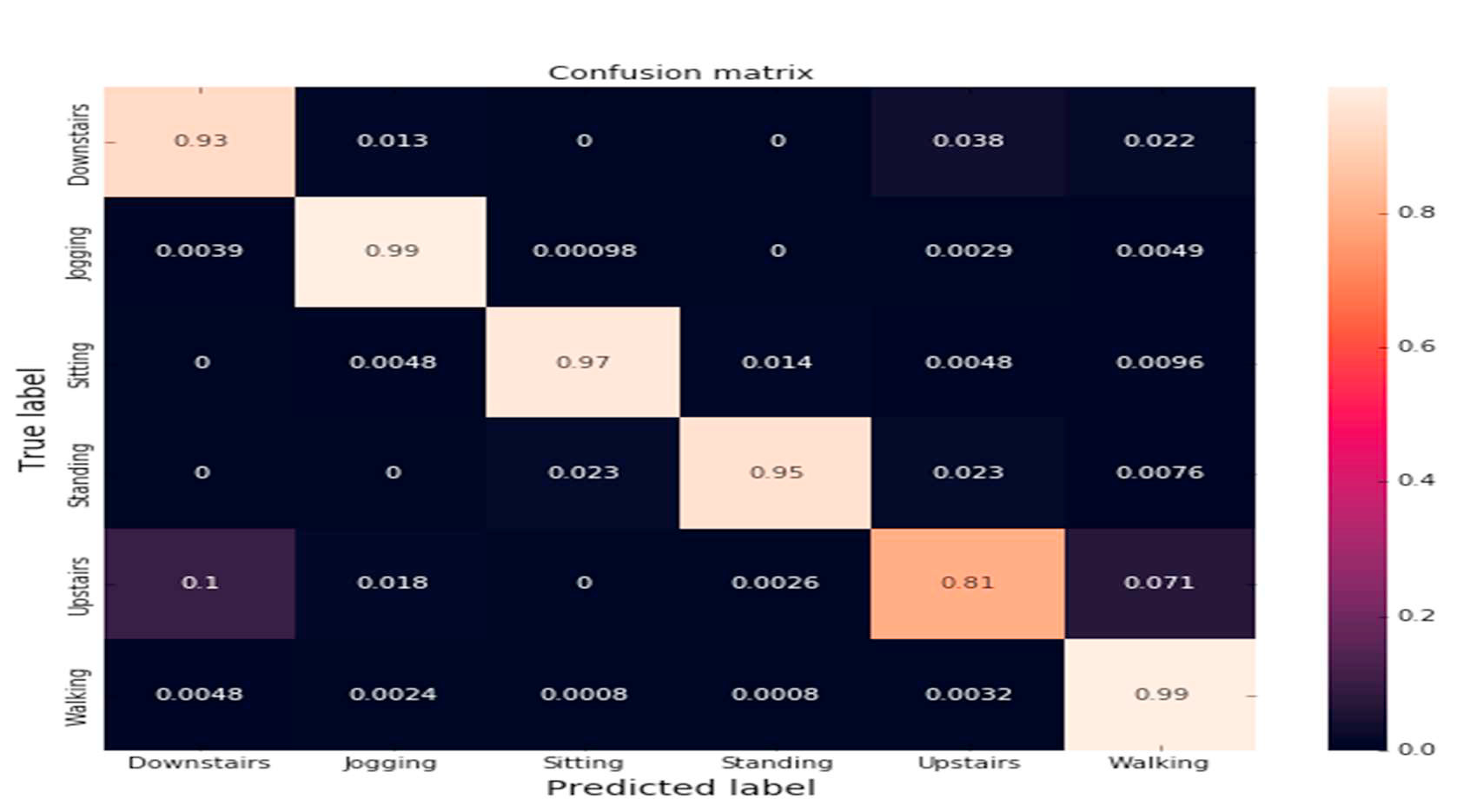
The below figure depicts the operation of a lightweight RNN-LSTM-based HAR system for edge devices. The reading from the accelerometer is divided into fixed windows of size T. A collection of readings (x1, x2, x3,.., xT-1, xT) recorded in time T, where xt is the reading recorded at any time instance t, serves as the input to the model. Readings from this segmented window are then fed into a lightweight RNN-LSTM model. The model combines the output from many states using a softmax classifier to create a single final output for that specific window as oT.



Wisdm dataset, split into 70:30 for training and testing, is used to train the lightweight RNN-LSTM. According to the activation function, the model's weights are updated. The cost function between the predicted labels and the ground truth is the mean cross-entropy. The Adam optimizer is employed to update model parameters and minimize the cost function. This model was developed on a Raspberry Pi 3 to test its edge device compatibility. Using the hit-and-trial method, different combinations of parameters, including the number of epochs, batch size, window size, and learning rate, were examined.

* Performance metrics include – Accuracy, Precision, Recall & F1 Score

Result: The evaluation findings for the Lightweight RNN-LSTM model are presented in this part, along with comparisons to some of the earlier efforts. The below figure depicts the model's confusion matrix. For walking and jogging exercises, the Lightweight RNN-LSTM obtained a 99 percent accuracy. For upstairs action, a minimum accuracy of 81 percent is attained.



Equations are used to determine accuracy, precision, recall, and f1-score. Accuracy, precision, recall, and F1-score for Lightweight RNN-LSTM were all 95.78 percent, 95.81 percent, and 95.73 percent, respectively. Recall and precision are calculated to validate performance since accuracy could generate false results if the data in each class of dataset is unbalanced.

Conclusion: In this paper, a lightweight HAR model is built. This model is implemented on a Raspberry Pi3 edge device. Communication delay, costs, and network traffic are all decreased when human activities are recorded on edge devices. In comparison to numerous other machine learning and deep learning models, the proposed model yields better outcomes

Future Work: This research can be expanded in the future to distinguish more intricate behaviors. It may be installed on different edge devices running iOS or Android. Static sliding window architecture was used in the development of this model. Future testing of this design can also include a dynamic windowing mechanism. A single tri-axial accelerometer was also used in the development of this system. Multi-sensor data can be supported by expanding it.