Human activity recognition from inertial sensor time-series using batch normalized deep LSTM recurrent networks

Tahmina Zebin, Matthew Sperrin, Niels Peek and Alexander J. Casson, *Senior Member, IEEE*

***Abstract*— In recent years machine learning methods for human activity recognition have been found very effective. These classify discriminative features generated from raw input sequences acquired from body-worn inertial sensors. However, it involves an explicit feature extraction stage from the raw data, and although human movements are encoded in a sequence of successive samples in time most state-of-the-art machine learning methods do not exploit the temporal correlations between input data samples. In this paper we present a Long-Short Term Memory (LSTM) deep recurrent neural network for the classification of six daily life activities from accelerometer and gyroscope data. Results show that our LSTM can processes featureless raw input signals, and achieves 92% average accuracy in a multi-class-scenario. Further, we show that this accuracy can be achieved with almost four times fewer training epochs by using a batch normalization approach.**

I. INTRODUCTION

Many applications in healthcare make use of wearable activity monitors such as the well known Fitbit for day-to-day activity tracking [1], [2]. However the accuracy of these systems is still highly debated [3], and there is a significant amount of ongoing work for improving the performance of Human Activity Recognition (HAR) from inertial sensors. Recently *deep learning* approaches to machine learning have gained a significant amount of research interest. However to date the application of deep learning models to train time-series of inertial sensor data for activity recognition is still under-explored [2], [4]–[8]. Deep learning models such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) employ a data-driven approach to learning discriminative features from raw sensor data to infer complex, sequential, and contextual information in a hierarchical manner [4], [9]. This avoids the need to perform an explicit feature generation and selection stage, which is time consuming when handcrafting features, can introduce bias with different feature sets being best suited to different data sets, and complicates direct comparisons of performance as different features are used in different studies. In addition, deep learning approaches are highly suited for exploiting temporal correlations in data sets. Developed for applications such as speech recognition, language modeling, and video processing, CNNs and RNNs can take contextual relationships in data sequences into account [2], [5].

They are thus very suitable for application to HAR classification where potentially a large amount of data is available, and human movements are encoded in a sequence of successive samples in time and the current activity is not defined by one small window of data alone. A number of recent studies have investigated this. [7] used CNNs to classify activities using data from multiple inertial sensors on the body. This performed well, and was optimized for low-power devices, but reintroduced the extraction of handcrafted features by using a spectrogram of the input data, and it required multiple sensor devices (whereas most people wear just one unit at any time). [5] and [10] used Long-Short Term Memory (LSTM) RNNs to better exploit the temporal correlations between input data samples by having memory cells and controlling gates embedded in the architecture. Hybrid models combining CNN and LSTM RNNs were also suggested in [7]. These have achieved a high level of classification accuracy. [5] used a *DeepConvLSTM* network and achieved 91.7% accuracy. [7] reported an F1 score of 90.8% with a factorized LSTM network.

However, to our knowledge no previous studies have used a stacked LSTM architecture, which has the capability of providing better generalization and robust temporal pattern learning [11], for HAR classification. Previous works are also focused on mostly accelerometer based datasets, not making use of the additional sensing available in modern internal sensor units such as gyroscopes. In this paper we have implemented and improved a stacked LTSM architecture for the feature-free classification of activities using both accelerometer and gyroscope signals as the raw data input. To make the network fast and robust we have employed dropout regularization and the recently introduced batch normalization method. Batch normalization has previously been demonstrated in feed-forward networks and has also found limited use in stacked RNNs for text and speech processing [12], where the normalization is applied to the input of each RNN only for very short sequences (10 samples). We successfully applied the technique to LSTMs with 128 sample sequences and found a significant reduction in training epochs in comparison to the generic LSTM model. To our knowledge, this is the first application of the method to HAR classification. We achieved a 92% average classification accuracy with a six-class inertial sensor dataset, and a four times reduction in the number of training epochs required.

The remainder of this paper is organized as follows.

This work was supported by the UK Engineering and Physical Sciences Research Council grant number EP/P010148/1.

T. Zebin and A. J. Casson are with the School of Electrical and Electronic Engineering, The University of Manchester, Manchester, UK. Email: {tahmina.zebin,alex.casson}@manchester.ac.uk.

M. Sperrin and N. Peek are with the Health e-Research Centre, The University of Manchester, Manchester, UK. Email: {matthew.sperrin,niels.peek}@manchester.ac.uk.

Section II introduces the mathematical background of the LSTM architecture. Section III gives details on our parameter settings for the implemented network and testing methods, and the classification performance results shown in Section IV.

II. MATHEMATICAL BACKGROUND

RNNs such as Long-Short Term Memory (LSTM) networks can learn very long-term dependencies [7], which makes them well suited to model temporal dynamics in activity time-series. They retain important data from the previous inputs and use that information to modify the current output. In this section we introduce the fundamental components and mathematical model of a generic LSTM unit, and our used batch normalization approach.

A. LSTM architecture

The fundamental LSTM unit is shown in Fig. 1, and is composed of a cell with an input gate, output gate, and forget gate. LSTMs use the concept of *gating* to deal with the vanishing or exploding gradient problem [11]. The cell is responsible for *remembering* values over arbitrary time intervals, and each of the three gates can be thought of as a conventional artificial neuron, computing an activation (using an activation function) of a weighted sum of the current data x_t, a hidden state h_{t-1} from the previous time step, and any bias b. Intuitively, the gates can be thought as regulators of the flow of values through the connections of the LSTM [5], [11]. At each time step they control which operation is performed by the cell as defined below. In (1) to (6), w_i are the weights associated with each multiplication at gate i, and σ and $\tanh$ are options for the activation functions.

In Fig. 1 the input gate controls the extent to which a new value flows into the cell, known as a *write* operation:

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i). \tag{1}$$

The forget gate performs a similar operation, controlling the extent to which the current cell value is kept, doing a *reset* operation:

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f). \tag{2}$$

The candidate memory cell is updated similarly as

$$\widetilde{C_t} = \tanh(w_c[h_{t-1}, x_t] + b_c) \tag{3}$$

and by combining these different internal values the internal long-term memory or the next cell memory is generated as:

$$c_t = f_t \circ c_t + i_t \circ \widetilde{C_t}. \tag{4}$$

From this, the cell output is generated by the output gate to control the extent to which the value in the cell is used to compute the output activation, doing a *read* operation:

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o). \tag{5}$$

Finally the cell's hidden output is found as

$$h_t = o_t \circ \tanh(c_t) \tag{6}$$

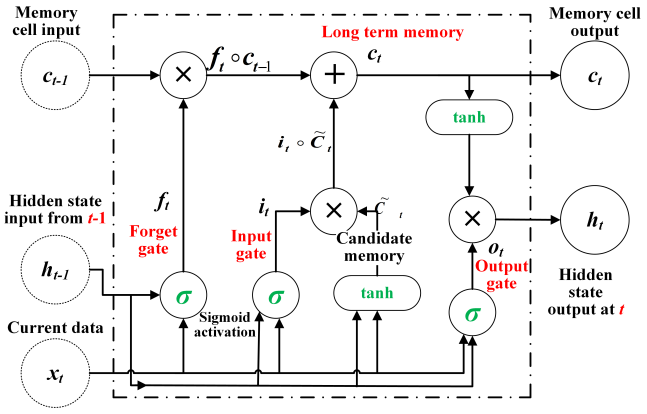

Fig. 1. Illustration of a Long-Short Term Memory (LSTM) unit.

for passing to other cells in the deep network. Each of the gates has parameters for its weights and biases, giving a large number of parameters for deep networks with many units present. The weights of these connections are learned or updated during the training of the network.

B. Batch normalization

Training LSTMs is complicated by the fact that the statistical distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it extremely hard to train models with saturating non-linearities [12]. Batch normalization has recently been introduced to overcome this by normalizing the x_t and h_{t-1} activations going into each layer by applying a covariate shift. This enforces the means and variances of x_t and h_{t-1} to be invariant to changes in the parameter distributions of the underlying layers and effectively decouples each layers parameters from those of other layers, leading to a better-conditioned optimization problem [12]. We have embedded the batch normalization technique in our proposed model discussed in the next section.

III. METHODS

A. Proposed LSTM model

A schematic diagram of our multi-layer stacked architecture LSTM network for multi-class HAR classification is presented in Fig. 2. The model architecture is novel in its use of longer temporal sequences in the LSTM and its use of batch normalization for HAR with the RNN architecture when compared to ones reported in the literature [8], [9].

As activity data is recorded from the sensor as a time-series, preparing the training data as per the requirements of the LSTM is crucial for building and training. In our LSTM implementation the data input x_t is multi-dimensional, containing three channels from a 3-axis accelerometer, and three from a 3-axis gyroscope. We needed to reshape these six parallel 1-D time series data into the 3-D structure required by an LSTM with the specific number of neurons in one dimension, the number of memory steps to process per time step in another dimension, and different sensor channels on

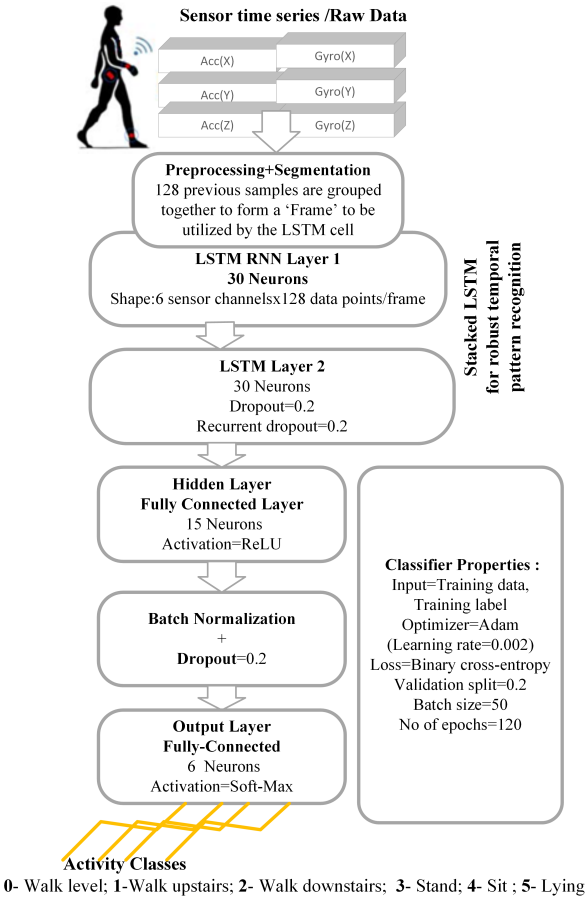

Fig. 2. Proposed deep LSTM network architecture for HAR classification.

the third dimension. For the architecture shown in Fig. 2 we prepared a reformatted matrix of shape: No. of data rows $\times$ 128 samples/sequence $\times$ 6 channels. In every data row 127 previous samples were arranged to work as a memory for the current data.

The model is implemented using the *Keras* open source library in Python [13], and we have utilized the *sequential* model and with Dense, LSTM, Dropout, and Batch Normalization layers. The input layer has 30 neurons using the 128 previous data points. A second LSTM layer was stacked in our model to utilize a deeper time dependency in predicting the next value. Finally, the network was converted into a classifier using a fully connected hidden Dense layer with 15 neurons followed by an output Dense layer of six neurons with a soft-max classifier to provide probabilistic assignments of labels/classes from the raw data. The final model was trained with an Adam optimizer with a learning rate of 0.002 and binary cross-entropy [13].The training of the model is done offline without any GPU, on a conventional computer with a 2.4GHZ CPU and 16GB memory.

B. Dataset Preparation

To evaluate the performance of the model, we processed the time series data from a waist mounted inertial sensor recorded at 50 Hz sampling frequency containing both accelerometer and gyroscope measurements. Data for 20 subjects is present, described in detail in [14]. The dataset

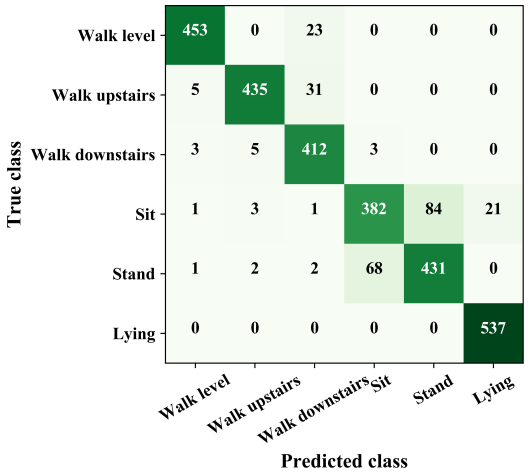

Fig. 3. Confusion matrix for the test set, and per-class sample numbers.

contains six everyday activities: 0–walk on level surface; 1–walk upstairs; 2–walk downstairs; 3–sitting; 4–standing; 5–lying. We kept the data from 14 volunteers, with approximately 7500 labeled activities as training data, and data from 6 volunteers, 3000 labeled activities, as a test dataset. The test set was separated entirely from the training dataset during our experiments. Also to avoid over-fitting the model with training data, 20% of the training dataset was held back as a validation set.

IV. RESULTS AND DISCUSSION

A. Performance Evaluation

To verify the performance of our LSTM model Fig. 3 shows the full confusion matrix of the test set. Some misclassifications are present, but overall the classification is highly accurate. This is quantified in Fig. 4 the via precision and F1 score. For activities such as walking level, walking up and walking down the average precision is over 95%. These time-dependent dynamic activities benefited from the LSTM memory processing for highly accurate classification. In contrast, from Fig. 3 and Fig. 4 it is clear that in our test data set most of the misclassifications are for static activities such as sitting and standing. These have less temporal correlations and repetitive components for learning by the LSTM.

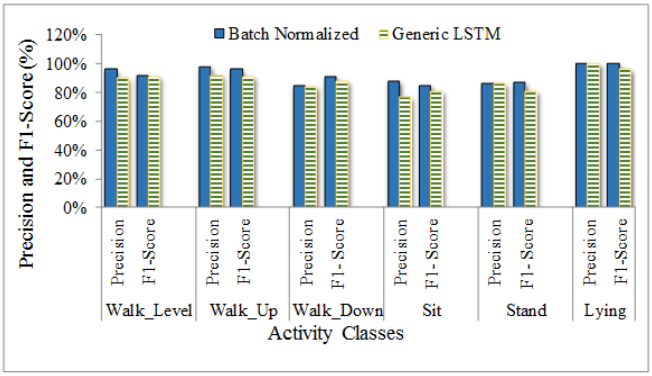

Fig. 4. Class-wise performance on the test data set assessed via precision and F1 score for a generic LSTM and a batch normalized LSTM.

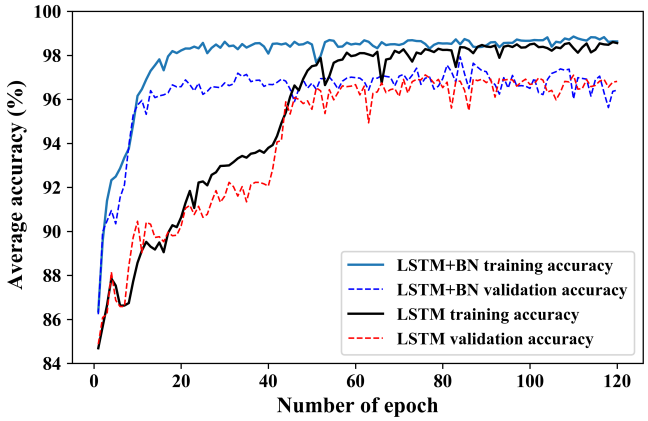

Fig. 5. Average training and validation set accuracy performance over 120 epochs for an LSTM model with and without Batch Normalization (BN).

TABLE I

QUANTITATIVE COMPARISON OF SVM, FULLY CONNECTED DENSE NEURAL NETWORK AND LSTMS FOR HAR CLASSIFICATIONS.

Learning method	Dataset	Accuracy (%)
SVM with handcrafted features [14]	[14]	93.4
Fully connected DNN [14]	[14]	91
DeepConvLSTM [5]	[15]	91.7
Factorized LSTM [7]	[15]	90.8
This work: LSTM (raw data in, no BN)	[14]	88
This work: LSTM+BN (raw data in)	[14]	92

B. Performance improvement with batch normalization

Fig. 4 also shows the LSTM performance when Batch Normalization (BN) is employed during training and validation of the network. The batch normalized LSTM (LSTM+BN) consistently outperforms the generic LSTM model with a class-wise accuracy of 92% (up 4% from the LSTM without BN). In addition, Fig. 5 plots the performance of the network training process for different numbers of iterations. During training the LSTM+BN achieves 98% training set accuracy four times faster (using 20 epochs) than the generic LSTM (80 epochs). This is also true with the validation data set. Potentially this allows a reduction in the training epochs required, and will be of vital importance for training future networks with bigger datasets.

C. Comparison with other approaches

To place our results in context Table I summaries the performance of other classification techniques when applied to the same data set as used in this paper, and to previous LSTM models. Our proposed LSTM+BN that processes featureless raw signals achieves 92% overall classification accuracy which is slightly lower than the Support Vector Machine (SVM) method in [14] which used handcrafted features to achieve 93.4% accuracy. This compromise in accuracy can be discounted by the fact that the LSTM classification is more generalized and capable detecting activities that have long term dependence which is not the case for SVM. Our LSTM performance is similar to that reported previously for LSTM activity classification, but the use of batch normalization can potentially obtain similar accuracy with substantially fewer training epochs.

V. CONCLUSIONS

We have presented an LSTM model with 92% average recognition accuracy for 6 daily-life activities using raw accelerometer and gyroscope data as the input. Dropout regularization and batch normalization made the network fast and robust in terms of speed and performance accuracy and achieved a four times reduction in training epochs required which will be beneficial when training on large amounts of data with a wide variety of complex activities. We will verify the effectiveness of our approach by testing it on a larger physical activity dataset from UK Biobank[16].

REFERENCES

[1] C. Torres-Huitzil and A. Alvarez-Landero, "Accelerometer-based human activity recognition in smartphones for healthcare services," in *Mobile Health*, S. Adibi, Ed., Cham: Springer, 2015, pp. 147–169.

[2] J. Wang, Y. Chen, S. Hao, *et al.*, "Deep learning for sensor-based activity recognition: A survey," *ArXiv preprint*, 2017, 1707.03502.

[3] C. K. Wong, H. M. Mentis, and R. Kuber, "The bit doesn't fit: Evaluation of a commercial activity-tracker at slower walking speeds," *Gait & Posture*, vol. 59, no. 1, pp. 177–181, 2018.

[4] A. Murad and J. Y. Pyun, "Deep recurrent neural networks for human activity recognition," *Sensors*, vol. 17, no. 11, p. 2556, 2017.

[5] F. J. Ordonez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[6] Y. Guan and T. Plotz, "Ensembles of deep LSTM learners for activity recognition using wearables," in *ACM Interact. Mob. Wearable Ubiquitous Technol.*, New York, Jun. 2017.

[7] N. Y. Hammerla, S. Halloran, and T. Plotz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *ACM IJCAI*, New York, Jul. 2016.

[8] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun. Surv. Tutor*, vol. 15, no. 3, pp. 1192–1209, 2012.

[9] H. Chen, J. Chen, R. Hu, *et al.*, "Action recognition with temporal scale-invariant deep learning framework," *China Commun.*, vol. 14, no. 2, pp. 163–172, 2012.

[10] D. Ravi, C. Wong, B. Lo, *et al.*, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," in *IEEE BSN*, San Francisco, Jun. 2016.

[11] K. Greff, R. K. Srivastava, J. Koutnik, *et al.*, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, 2017.

[12] C. Laurent, G. Pereyra, P. Brakel, *et al.*, "Batch normalized recurrent neural networks," in *IEEE ICASSP*, Shanghai, Mar. 2016.

[13] F. Chollet. (2013). Keras: The python deep learning library, [Online]. Available: `https://keras.io/`.

[14] T. Zebin, P. J. Scully, and K. B. Ozanyan, "Evaluation of supervised classification algorithms for human activity recognition with inertial sensors," in *IEEE Sensors conf.*, Glasgow, Nov. 2017.

[15] R. Chavarriaga, H. Sagha, A. Calatroni, *et al.*, "The opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 2033–2042, 2013.

[16] A. Doherty, D. Jackson, N. Hammerla, *et al.*, "Large scale population assessment of physical activity using wrist worn accelerometers: The UK biobank study," *PLoS One*, vol. 12, no. 2, 2017.