Bidirectional Gated Recurrent Units For Human Activity Recognition Using Accelerometer Data

Tamam Alsarhan, Luay Alawneh, Mohammad Al-Zinati, Mahmoud Al-Ayyoub
Jordan University of Science & Technology
thalsarhan14@cit.just.edu.jo, lmalawneh@just.edu.jo, mhzinati@just.edu.jo, maalshbool@just.edu.jo

Abstract—**Human activity recognition aims to detect the type of human movement based on sensor data gathered during human activity. Time series classification using deep learning approaches offers opportunities to avoid intensive handcrafted feature extraction techniques where the efficiency and the accuracy are heavily dependent on the quality of variables defined by domain experts. In this paper, we apply recurrent neural networks on data collected from mobile phone accelerometers for the recognition of human activity. More specifically, we use the bidirectional gated recurrent units mechanism. The results show that this technique is promising and provides high quality recognition results.**

Keywords—Mobile Sensors, Recurrent Neural Networks (RNN), Long-Short Term Memory (LSTM), Classification

I. Introduction

The basic objective of human activity recognition is to identify the actions of humans to assist them accordingly. Several studies targeted human activity recognition using various methods mainly for healthcare purposes [1-3]. With the explosion of data and the significant advancement in modern artificial intelligence technologies, the scale of data-centered applications in human society has grown rapidly. Human activity recognition is one of the modern artificial intelligence applications, which is mainly a classification problem. This field has recently gained attention due to its application in business management, surveillance and security, transport and logistics, medical care, eldercare, traffic management, and performance analysis in sports [4].

A wide variety of classical approaches have been used for classification. However, the problem with these approaches is that they rely on heuristic handcrafted feature extraction methods which affect their generalization. Deep learning approaches emerged to eliminate these constraints as they succeed in dealing with time sequential data that embodies correlations between close data points in a sequence [5]. Recent studies proposed to use Gated Recurrent Units (GRU) [5], a gating mechanism in recurrent neural networks (RNN), for classification. The GRU resembles Long-Short Term Memory (LSTM) with a forget gate having fewer number of parameters. It is a promising recurrent architecture that is flexible and capable of incorporating context information in case of sequence learning. Additionally, it succeeds in dealing with time series data that embodies correlations between data points that are close in the sequence.

The purpose of this study is to investigate whether RNN with GRU-cells can be used to effectively recognize human activities using accelerometer data. Using RNN for time series classification can be challenging as performance is highly dependent on the availability of trained data. Hence, we propose a bidirectional GRU (BiGRU) model for the classification of several daily life activities and fall states. The model uses the UniMiB SHAR open source dataset [6] for training and evaluation.

The rest of the paper is organized as follows. In Section II, we present a number of related studies. In Section III, we present our approach followed by an evaluation in Section IV. Finally, we conclude our paper in Section V with a highlight on future direction.

II. Related Work

Human activity recognition relies on feature extraction procedures using supervised classification techniques [7-10]. An early study by Foerster et al. [11] demonstrated the reliability of utilizing accelerometer data for human activity recognition. Ravi et al. [9] proposed a deep learning approach where they combined extracted features learned from the gathered sensor data with complementary information from a range of shallow features. This combination led to accurate and real-time activity classification.

Hassan et al. [10] used Deep belief network (DBN) to identify human activities using inertial sensor-based approach in smart phones. The results outperformed traditional approaches such as SVM and ANN. Jiang et al. [12] applied a Deep Convolutional Neural Network (DCCN) model to study the human activity recognition by taking the signal sequences of accelerometers and gyroscopes, and assembled them into a novel activity image. The DCCN model was able to automatically extract and learn the optimal features from the activity image for the recognition task. Karen et al. [13] proposed two DCCN architectures and used Softmax to reduce the overfitting. Similarly, Torfi et al. [14] applied 3D CNN that used both spatial and temporal dimensions for feature extraction and performed subsampling and convolution in different channels.

LSTM neural networks deal with the vanishing gradient problem [15, 16]. It is capable of handling complex serial information with long dependencies since it utilizes gating scheme for data representation. Thus, it is a strong candidate for studying time series data. Mehdiyev et al. [17] proposed stack LSTM Autoencoder network. This model extracts features in an unsupervised or self-supervised manner using the principles of deep learning. The time series data obtained from stack LSTM Autoencoder networks were passed to feed the forward neural network for classification. The neural network consists of three hidden layers where each layer consists of 200 neurons. They trained the network about 100 epochs and changed the training parameters such as activation functions, number of layers and neurons, and the type of optimization function. Finally, they used the average accuracy and recall as classification metrics.

Graves et al. [18] proposed the bidirectional long short-term memory (BLSTM) which uses the concept of incremental learning and handles long-range contextual processing. They used two large unconstrained handwriting databases. The proposed approach achieved a recognition accuracy of 79.7%. Veeriah et al. [19] proposed differential gating scheme for the LSTM neural network (dRNN). They used the KTH 2D action recognition and the MSR Action3D datasets. The results showed that the dRNN model outperformed the conventional LSTM approach.

Recurrent neural network (RNN) is a dynamic structure that makes efficient use of temporal information of the input sequence. Husken et al. [20] used dynamic recurrent neural

978-1-7281-1634-1/19/$31.00 ©2019 IEEE

network for time series classification. The time series fed into the network was represented by the output of the classification neurons. The results showed that the inclusion of a prediction task during learning strongly supports the learning process. Moreover, the generalization ability was significantly improved.

Recurrent neural networks (RNN) and LSTM Models were combined to perform different types of classification. Chung et al. [21] suggested RNN and LSTM models for Lexical Utterance Classification. The results indicated that the proposed approach outperformed the ngram-based language models (LMs), feed forward neural network LMs, and boosting classifiers. Furthermore, RNN worked well for short utterance series while LSTM worked efficiently for long series.

Finally, studies showed that Gated Recurrent Units (GRU) and the LSTM model performed well with long sequence applications [22, 23]. Vu et al. [24] proposed to use Self-Gated RNN for human activity recognition on data gathered from wearable sensors. The results showed that their approach outperformed standard RNN and maintained comparable results to those of LSTM and GRU. In this paper, we demonstrate the efficacy of using Bidirectional GRUs on time series data for human activity recognition.

III. Approach

We study the usefulness of the Gated Recurrent Unit (GRU) model in recognizing human activities of data gathered from mobile phone sensors. GRU is a variation of LSTM as they share a common design. Moreover, similar to LSTM, GRU adaptively resets or updates its memory content. Thus, each GRU has a *reset* gate and an *update* gate, which are analogous to the *forget* gate and the *input* gate of the LSTM. Additionally, in some cases, GRU and LSTM produce equally excellent results. However, GRU is different from LSTM in that it fully exposes its memory content each time step. Moreover, GRU strictly balances between the previous memory content and the new memory content using leaky integration.

The activity recognition task is a multiclass classification problem. Thus, we classify the activity samples into one specific class among different candidate classes. Activities are recorded as time series data using accelerometer sensors. Given a series of triaxial accelerometer data, the GRU model is supposed to generate hypotheses $\check{y}$ of the actual set of labels y. The power of the GRU is that it can handle long term dependencies. Hence, it has the ability to remove or add information to the cell state, carefully regulated by the gating scheme. Thus, GRU cells are able to remember important information about the received input. This enables them to be very precise in predicting next input.

Our proposed model consists of two distinct GRU layers that split the state neurons of a regular GRU in two parts. The first part is responsible for the positive time direction (forward states) and the second part is responsible for the negative time direction (backward states).

We divided the data into training and testing sets (80% for training and 20% for testing [25]). Our model determines the first 80% of the samples as training data while the remaining 20% as testing data. Additionally, we randomized the data using shuffling to ensure that the training and the testing sets contain all the possible cases defining the problem. The data includes 453-dimensional patterns obtained by concatenating the 151 acceleration values (x, y, z) recorded along each Cartesian direction.

We placed two independent GRU models together and the result is BiGRU. Since the model is bidirectional, forward and backward cells were used. Figure 1 shows that both layers are independent except that they share the same input sequence (X_1-X_4) also the final outputs (O_1-O_4) from the two layers are concatenated.

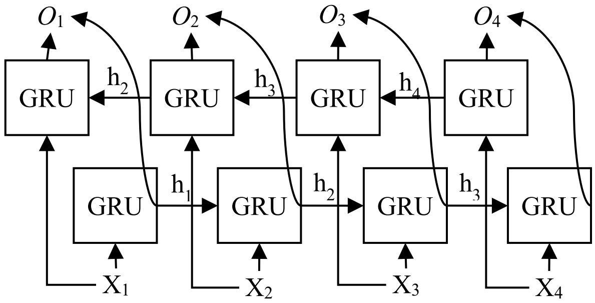

Figure 1. BiGRU Structure

In this model, forward and backward information about the sequence at each time step are obtained since we are using the two layers. This improves the learning process since the neighboring data-pairs (in time) are intuitively somehow dependent. The design of GRU helps to exploit useful information from the previous data point in the sequence. On the other hand, future input information coming up later in the sequence is usually useful for prediction. Thus, the learning performance will improve when using bidirectional structure. In [26], it was shown that the Bidirectional RNN structure can acquire better results than the other artificial neural network structures in case of time series data. Our BiGRU runs the inputs in two ways. One way is from the past to the future and the other is from the future to the past.

In classification problems, it is mandatory to define a loss function to describe the accuracy of the model's solution to a specific problem. Generally, the smaller the loss is, the smaller the deviation between the result of the model and the true value. In our model, we used the log loss function (known as the cross entropy loss) as a cost function. The cost function is a measure of how inaccurate the model is, by determining the ability of the model to find a relationship between input data and output labels.

For optimizing the model, we chose the Adam optimization function for the training [27]. This optimizer is an adaptive version of the stochastic gradient descent. Actually, the best optimizer is the one that produces the best and fastest results when updating the network parameters (weights and biases).

Since we have 11771 examples, overfitting is a considerable obstacle, which is also a common problem in machine learning. If the model is over-fitted, the resulting model is almost useless. This problem is observed when the loss function of the model is small in the training data while it is large in the testing data. This causes a large difference between the training accuracy and the testing accuracy. Our model tries to restrain the overfitting by using the *Dropout* function which is a formalization method that is widely used in modern deep learning environments [28]. It randomly selects some of the cells (neurons) in the neural layer and temporarily hides them. Then, it performs the training and the optimization process of the neural network in a certain loop. In the next loop, it will hide some other cells (neurons) until the end of the training. Thus, it randomly skips some neurons at training in order to force the other neurons to pick up the slack. As our model is bidirectional, we applied the *Dropout* in both GRU layers. *Dropout* is a clever way for regularization. It reduces the network tendency to become over-dependent on some neurons as they may not be available all the time.

We tuned the hyper-parameters in order to discover the values that acquire the best prediction results. We used the following training hyper-parameters for classification.

1. *Number of Epochs*: an epoch means passing the entire dataset forward and backward through the model only *once*.
2. *Number of iterations*: number of batches needed to complete one epoch.
3. *Batch size*: number of training examples present in a single batch.
4. *Optimizer function*: help minimize the error function in a specific model.
5. *Hidden units*: hidden neurons in RNN structure.
6. *Learning rate*: how fast weights are changed.

The training process of the model contains the following steps.

- Input the dataset samples into the BiGRU model.
- Initialize the learning parameters: weights and biases (randomly at the first time).
- Compute the cell states of the model using the inputs and the learning parameters.
- Compute the prediction of the model $\check{y}$.
- Compute the cross entropy loss.
- Train the network using *Adam* optimizer: adjust weights and biases based on the computed loss.
- Repeat until reaching the highest accuracy.

We used the training accuracy and the testing accuracy as a measure of recognition. Accuracy is measured using Equation 1.

$$accuracy = \frac{\text{The amount of correct classifications}}{\text{The total amount of classifications}} \quad (1)$$

The training accuracy is the accuracy calculated when the model is applied to the training data while the testing accuracy is the accuracy calculated when the model is applied to the testing data. Both accuracies are important to identify the overfitting.

IV. Evaluation

We implemented our model using Tensorflow, an open-source software library, for dataflow programming across a range of tasks [29]. We created three virtual machines using Google Cloud Platform to start training the model on the UniMiB SHAR dataset [6] using 24 vCPUs with 125GB of Memory. The dataset contains 11771 data samples for daily life activities, and fall states acquired using smart phones. Data collection is performed by 30 persons (24 females and 6 males) ranging from 18 to 60 years old [6]. The different characteristics of the subjects in the UniMiB SHAR dataset are shown in Table 1.

Samsung Galaxy Nexus I9250 with the Android OS version 5.1.1 was used in the experiments that aimed to build the dataset using its Bosh BMA220 acceleration sensor. This triaxial low-g acceleration sensor allows measurements of acceleration in three perpendicular axes (*x*, *y*, and *z*). For each activity, the accelerometer data vector is made of three vectors of 151 values (total vector size is 151 x 3 = 453), one for each acceleration direction. Overall, the main feature used from this dataset is the acceleration value in the *x*, *y*, *z* axes.

Table 1. The Characteristics of Subjects

	Female	**Male**	**Total/Range**
Subjects	24	6	30
Age	18-55	20-60	18-60
Height (cm)	160-172	170-190	160-190
Weight (kg)	50-78	55-82	50-82

Samples are divided into 17 fine grained classes and grouped in two coarse grained classes. The first class (*A*) contains samples of nine types of activities of daily living (*ADL*) and the second class (*F*) contains samples of eight types of falls. The dataset contains 7579 ADL states and 4192 fall states.

We used four different classification tasks for the evaluation (*AF-2*, *F-8*, *A-9*, and *AF-17*) as follows.

AF-2 consists of two classes obtained by considering all the ADLs as one class and all the fall states as another class. It allows evaluating the classifier's robustness in distinguishing between ADLs and fall states.

F-8 consists of eight classes obtained by considering all the classes of fall states. It allows evaluating the ability of the classifier to differentiate among different types of fall states.

A-9 consists of nine ADL classes. It determines the ability of the classifier to distinguish among different types of ADLs.

AF-17 consists of 17 classes (nine classes of ADLs and eight classes of fall states). It evaluates the ability of the classifier to determine the type of move regardless of being an ADL or a fall state.

Table 2 shows the results obtained from running the BiGRU model on the raw data for the four tasks along with the values of the hyper-parameters. These results were achieved after several experiments and tuning of the parameters. For the binary classification (AF-2), the accuracy was very high (98.93%). However, we only reached 85.36% accuracy in case of F-8. The misclassification usually occurs in some fall states due to the similarity between some pairs such as Syncope and Falling leftward, Generic falling backward and Falling backward-sitting-chair. For A-9 and AF-17, the accuracy was up to 93.9%. Finally, training the AF-17 reached 10 hours since it has the largest training set among all the other tasks.

Table 2. Accuracy of the Model

Parameter	**AF-2**	**F-8**	**A-9**	**AF-17**
Epochs	50	100	70	50
Iterations	45	13	12	148
Batch size	200	256	512	64
Learning rate	0.001	0.001	0.001	0.001
Hidden units	200	200	200	200
Dropout	0.5	0.5	0.5	0.5
Accuracy	98.93%	85.36%	93.79%	93.9%

V. Conclusion

In this paper, we presented an approach for using bidirectional GRU model for human activity recognition. The UniMiB SHAR dataset was used for training and evaluating the model. The results showed that using the bidirectional GRU model is very effective in recognizing human activities gathered as time series data. In the future, we intend to compare the GRU model to the LSTM and the Vanilla RNN models. Moreover, we intend to experiment the effect of data augmentation on the accuracy of the model.

REFERENCES

[1] V. Osmani, S. Balasubramaniam, D. Botvich, "Human activity recognition in pervasive health-care: Supporting efficient remote collaboration," Journal of Network and Computer Applications, 31(4), 2008, pp. 628–655.

[2] P. Woznowski, R. King, W. Harwin, I. Craddock, "A human activity recognition framework for healthcare applications: ontology, labelling strategies, and best practice," In Proceedings of the International Conference on Internet of Things and Big Data (IoTBD) INSTICC, 2016, pp. 369–377.

[3] N. Bidargaddi, A. Sarela, L. Klingbeil, M. Karunanithi, "Detecting walking activity in cardiac rehabilitation by using accelerometer," Proc. 3rd Int. Conf. Intell. Sensors Sensor Netw. Inf., pp. 555-560, Dec. 2007.

[4] J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, "Deep learning for sensor-based activity recognition: A survey," Pattern Recognition Letters, 2018.

[5] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Holger, Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," 10.3115/v1/D14-1179.

[6] D. Micucci, M. Mobilio, P. Napoletano, "UniMiB SHAR: A new dataset for human activity recognition using acceleration data from smartphones," IEEE Sens. Lett. 2016, 2, 15–18

[7] M. Uddin, M. Hassan, A. Almogren, M. Zuair, G. Fortino, J. Torresen, "A facial expression recognition system using robust face features from depth videos and deep learning," Comput. Electr. Eng., 2017. https://doi.org/10.1016/j.compeleceng.2017.04.019.

[8] G. Hinton, S. Osindero, Y. Teh, "A fast learning algorithm for deep belief nets," Neural Comput. 18(7):1527–1554, 2006.

[9] D. Ravi, C. Wong, B. Lo, G. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," IEEE Journal of Biomedical and Health Informatics. 21(1): 56–64, 2017.

[10] M. Hassan, M. Uddin, A. Mohamed, A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," Future. Gener. Comput. Syst. 81:307–313, 2018.

[11] F. Foerster, M. Smeja, and J. Fahrenberg, "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring," Computers in Human Behavior, vol. 15,no. 5, pp. 571–583, 1999.

[12] W. Jiang, Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," In Proceedings of the 23rd Annual ACM Conference on Multimedia Conference. pp. 1307–1310. ACM (2015).

[13] K. Simonyan, A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," NIPS, 2014. arXiv:1406.2199v2.

[14] A. Torfi, S. Iranmanesh, N. Nasrabadi, J. Dawson, "3D Convolutional Neural Networks for Cross Audio-Visual Matching Recognition," IEEE Access 5: 22081-22091 (2017).

[15] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, K. Saenko, "Translating videos to natural language using deep recurrent neural networks," arXiv preprint arXiv:1412.4729, 2014.

[16] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, G. Toderici, "Beyond short snippets: Deep networks for video classification," arXiv preprint arXiv:1503.08909, 2015.

[17] N. Mehdiyev, J. Lahann, A. Emrich, D. Enke, P. Fettke, P. Loos, "Time Series Classification using Deep Learning for Process Planning: A Case from the Process Industry," Procedia Comput. Sci. 2017, 114, 242–249.

[18] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A Novel Connectionist System for Unconstrained Handwriting Recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 5, pp. 855-868, 2009.

[19] V. Veeriah, N. Zhuang, G.-J. Qi, "Differential recurrent neural networks for action recognition," In Proceedings of IEEE International Conference on Computer Vision (ICCV), pages 4041–4049, 2015.

[20] M. Husken, P. Stagge, "Recurrent neural networks for time series classification," Neurocomputing, 50 (2003), pp. 223-235

[21] S. Ravuri, A. Stolcke, "Recurrent neural network and LSTM models for lexical utterance classification," In INTERSPEECH-2015, 135-139.

[22] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," arXiv preprint arXiv:1412.3555, 2014.

[23] F. Gers, N. Schraudolph, J. Schmidhuber, "Learning Precise Timing with LSTM Recurrent Networks," Journal of Machine Learning Research, 3:115–143, 2002.

[24] T. Vu, A. Dang, L. Dung, J-C. Wang, "Self-Gated Recurrent Neural Networks for Human Activity Recognition on Wearable Devices," In Proceedings of the on Thematic Workshops of ACM Multimedia 2017 (Thematic Workshops '17). ACM, New York, NY, USA, 179-185

[25] K. Dobbin and R. Simon, "Optimally splitting cases for training and testing high dimensional classifiers," BMC Medical Genomics, vol. 4, no. 1, p. 31, 2011.

[26] M. Schuster, K. Paliwal, "Bidirectional recurrent neural networks," Signal Processing, IEEE Transactions on, 45(11), 2673–2681.

[27] D. Kingma, J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," J. Machine Learning Res. 15, 1929–1958 (2014).

[29] Tensorflow, https://www.tensorflow.org/