Proceedings of the 38th Chinese Control Conference
July 27-30, 2019, Guangzhou, China

Attention-based LSTM Network for Wearable Human Activity Recognition

Bo Sun[1,2], Meiqin Liu[1,2], Ronghao Zheng[2], Senlin Zhang[2]

1. State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China

2. College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China
E-mail: {sunboim, liumeiqin, rzheng, slzhang}@zju.edu.cn

Abstract: Sensor-based human activity recognition (HAR) has become a popular research topic because of its wide applications. Conventional machine learning approaches have made enormous progress in the past years. However, those methods rely on handcrafted features that are incapable of handling complex activities, especially with high dimensional sensor data. Deep learning technology, together with its various models, is one of the most accurate methods of working on activity data. In this paper, we propose an attention-based Long Short Term Memory (LSTM) network for wearable human activity recognition. Specifically, we construct an LSTM network to model the sensor readings, which has been proved to be very effective for time sequences. Then, we introduce the attention mechanism for the base LSTM network to learn which parts of the raw sensor data are more important for determining the overall activities. When tested with the Opportunity data set, the F1-score is increased by 2.6%, compared with baseline LSTM results.

Key Words: Human activity recognition, Body-worn sensors, Long short term memory, Attention mechanism

1 Introduction

Human activity recognition (HAR) is an important area of research in ubiquitous computing, human behavior analysis and human-computer interaction. Indeed, HAR is of value in both theoretical research and actual practice. It can be used widely, including in health monitoring [1], smart homes [2], and human-computer interactions [3]. Some simple activity-aware systems are now commercial in the form of fitness tracker or fall detection devices [4]. However, many scenarios of high societal value such as enabling natural human-robot interaction in everyday settings or complex activities of daily living (ADL), e.g.,cooking, washing up and sweeping the floor are still elusive. So far, there are mainly two types of HAR: video-based HAR and sensor-based HAR [5]. Video-based HAR analyzes videos or images containing human motions from the camera, while sensor-based HAR focuses on the motion data from smart sensors such as accelerometer, gyroscope, Bluetooth, light sensors and so on. Although video-based recognition method excels other recognition methods in indoor activity, it has several restrictions such as space limitation and interference from the environment [6]. With the development of sensor technology and computing power, the sensor-based HAR is becoming more promising with privacy well protected.

Our study is based on the hypothesis that specific human body movements translate into characteristic sensor signal patterns, which can be perceived and classified using pattern recognition techniques. Conventional pattern recognition approaches have made enormous progress on HAR by adopting algorithms such as naive Bayes, decision tree, support vector machine, and hidden Markov models [7]. However, in most daily HAR tasks, those methods may heavily rely on heuristic handcrafted feature extraction, which is usually limited by human domain knowledge [8]. Different from traditional machine learning methods, deep learning can largely improve the efficiency of extracting features and can learn much more high-level and profound features by training an end-to-end neural network. Among those methods, the Long Short Term Memory (LSTM) network which can learn infinite temporal contexts exhibits better performance in handling time sequences problems. However, it is not reasonable to assume that an event in the distant past would actually influence current events as there is typically only little relation between current and distant past activities [9].

In this paper, we describe an LSTM network with attention mechanism, which can automatically focus on the time series that has a decisive effect on classification, to capture the most important temporal dependencies from the input, without using extra handcrafted features and human domain knowledge. Attention mechanism allows a model to learn a set of weights over raw sensor data, which we leverage to weight the temporal context. We conduct experiments on the Opportunity data set [10], and achieve an F1-score of 90.9%, higher than baseline LSTM.

The rest of this paper is structured as follows. In Section 2, we review related works about the HAR problem. Section 3 presents our Attention based LSTM model in detail. Section 4 gives a detailed description of the setup of experimental evaluation and the experimental results. Finally, we have our conclusion in Section 5.

2 Related Works

HAR aims to identify the actions carried out by a person given a set of observations of him/herself and the surrounding environment [11]. There is an abstract definition of sensor-based HAR described as follows [12]:
Suppose a user is going through some sorts of activities belonging to a predefined activity set A:

$$A = \{A_i\}_{i=1}^{m} \tag{1}$$

This work was supported by the Joint Fund of Ministry of Education for Pre-research of Equipment under Grant 6141A020223, and the "13th Five-Year Plan" Equipment Pre-Research Fund under Grant 61403120107.

where m is the number of activity types. There is a sequence of sensor reading that captures the activity information:

$$s = \{d_1, d_2, \cdots, d_t, \cdots, d_n\} \tag{2}$$

where d_t is the sensor reading at time t and n is the length of the sequence. What we need to do is to build a mapping F to predict the activity sequence based on sensor reading s:

$$\hat{A} = \left\{\hat{A}_j\right\}_{j=1}^{n} = F(s), \qquad \hat{A} \subset A \tag{3}$$

and true activity sequence (ground truth) is denoted as:

$$A^* = \left\{A_j^*\right\}_{j=1}^{n}, \qquad A^* \subset A \tag{4}$$

The target of HAR is to get a model F which can forecast activity sequence with high accuracy by minimizing the discrepancy between predicted activity $\hat{A}$ and the ground truth activity A^*.

Over the years, various methods have been proposed for HAR. Most of them are based on supervised learning and apply extra human domain knowledge to derive a set of high-level features. Mantyjarvi was the first one who used Principal Component Analysis (PCA) and wavelet transform to extract features from raw sensor data, and employed multilayer perceptrons in simple human activity (standing, up and down stairs, walking) recognition [13]. Olguın used the Hidden Markov Model (HMM) model as a classification model to compare the effects of different sensor locations on the final classification results. Olguın'work drew the conclusion that increasing the number of sensors can improve the classification accuracy [14]. In 2011, Kwapise and her colleagues utilized only one accelerometer embedded in a smartphone to recognize human activities. The results of their work revealed that compared with former machine learning techniques their J48 decision trees and multilayer perceptrons could reach higher performance [15]. However, a limitation of these works is the feature extraction and domain expertise required to analyze the raw data. This expertise would be necessary for each new data set or sensor modality. In essence, it is expensive and not scalable. Ideally, learning approaches could be used that automatically explore the features required to make accurate predictions from the raw data directly.

Recently, deep neural network models, especially Convolutional neural networks (CNN) and Recurrent neural networks (RNN) have started delivering on their promises of feature learning and are achieving state-of-the-art results for human activity recognition. Ronao and Cho used the CNN to automatically extract robust features and mine the intrinsic link of human activities and achieved accuracy of 95.75% in the identification of 6 different daily activities [16]. Hammer and Halloran evaluated the performance of DNN, CNN, and RNN through 4,000 experiments on some public HAR data sets and drew some conclusions that RNN and LSTM are recommended to recognize short duration activities that have natural order while CNN is better at inferring long-term repetitive activities [17]. However, these approaches assign the same weight of attention to different temporal contexts which might not be suitable for behavior sets with individually varying durations.

3 Methods

In this section we propose our attention-based LSTM (Att-LSTM) model in detail. As shown in Fig.1, the model contains four components:

(1) Input layer: input sensor data to this model;

(2) LSTM layer: utilize LSTM to get high-level features;

(3) Attention layer: produce a weight vector, and merge features from each time step into a temporal feature vector to find relevant temporal context by multiplying the weight vector;

(4) Output layer: the temporal feature is finally used for activity recognition. These components will be presented in detail in this section.

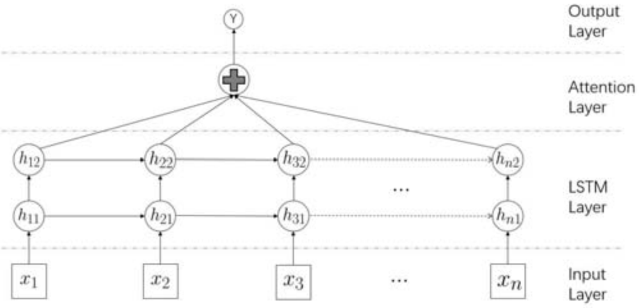


Fig. 1: LSTM with Attention

3.1 Channels and Sliding Window

LSTM network suitable for time series classification requires that the data is prepared in a specific manner in order to fit a model. There are mainly two important aspects. The first one is channel. Employing multi-channels enhances the representation capability of the model as it can reflect the hidden information of sensor inputs. Thus, we treat each axis of each sensor as separate channels. The second aspect is that the input should be cut into individual inputs based on the sampling rate. In Fig.2, we use a sliding window of fixed length n to segment the data of d channels and the last time step of the time window is chosen as a label for each sample's classification. According to Banos and Galvez's work, it is common to use one to two seconds of sensor data in order to classify a current fragment of activity [18]. Thus, based on the sample rate the window length is set to 1.44s to form the new next samples, using 50% overlap, which will capture the transition of one activity to another. Finally, the shape of the input is converted to $sampleNum \times windowLength \times channelNum$ with overlap.

3.2 LSTM Network

LSTM units were firstly proposed by Hochreiter and Schemidhuber to overcome gradient vanishing or exploding problems [19]. The main idea is to introduce an adaptive gating mechanism, which can ease the learning of temporal relationships on long time scales. Fig.3 illustrates the architecture of a standard LSTM cell.

Formally speaking, each cell in LSTM can be computed

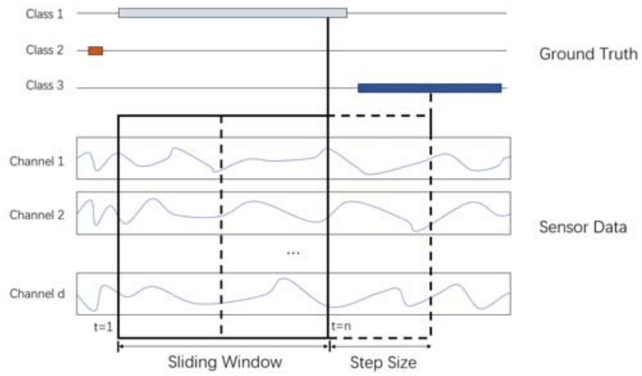


Fig. 2: Using a sliding window to segment the channels' data

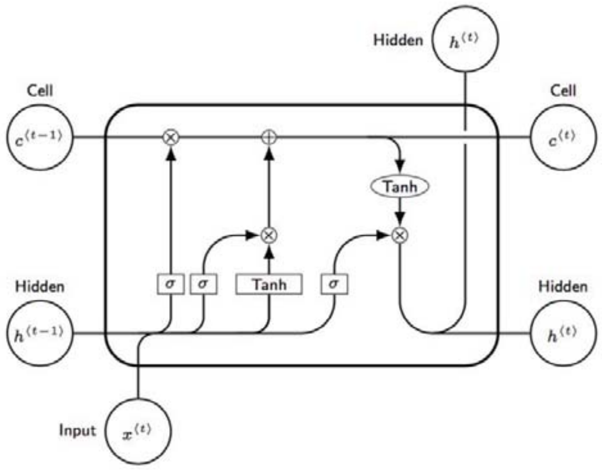


Fig. 3: A LSTM cell

as follows:

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}^T, \qquad h_{t-1} \in \mathbb{R}^m, x_t \in \mathbb{R}^n \tag{5}$$

$$f_t = \sigma(W_f \cdot X + b_f) \tag{6}$$

$$i_t = \sigma(W_i \cdot X + b_i) \tag{7}$$

$$o_t = \sigma(W_o \cdot X + b_o) \tag{8}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c \cdot X + b_c) \tag{9}$$

$$h_t = o_t \odot \tanh(c_t) \tag{10}$$

where $[h_{t-1}; x_t] \in \mathbb{R}^{m+n}$ is a concatenation of the previous hidden state h_{t-1} and the current input x_t (m is the size of hidden state, n is the length of inputs), $W_i, W_f, W_o \in \mathbb{R}^{m\times(m+n)}$ denote the weighted matrices, $b_i, b_f, b_o \in \mathbb{R}^m$ denote the biases of LSTM network to be adjusted during training, parameterizing the transformations of the input, forget and output gates respectively, $\odot$ denotes element-wise multiplication, and σ stands for the logistic sigmoid function.

3.3 Attention

Attention mechanism has recently shown great success in a wide range of tasks ranging from machine translations, speech recognition to dialogue systems [20][21][22]. An attention function can be described as mapping a query and a set of key-value pairs to an output. As shown in Fig.4, there are main three steps in calculating the attention value:

(1) The first step is to calculate the similarity between the query and each key to get the weights;

(2) The second step is to normalize these weights using a softmax function;

(3) Finally, the weights and the corresponding values are weighted and summed to obtain the final attention value;

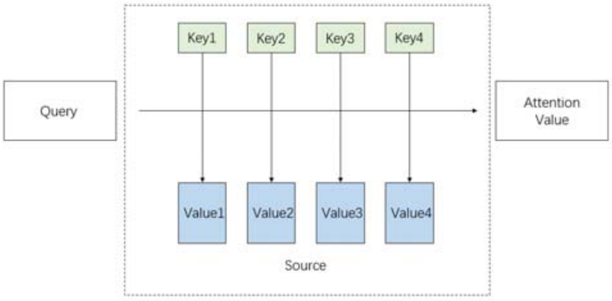


Fig. 4: Attention function

Instead of encoding the input sequence into a single fixed context vector, the attention mechanism can allocate attention and learn inner relationship, by adjusting the weights they assign to various inputs. It distributes attention over several hidden states, illustrated in Fig.5 in the lines of different thicknesses and color (red denotes the valid states).

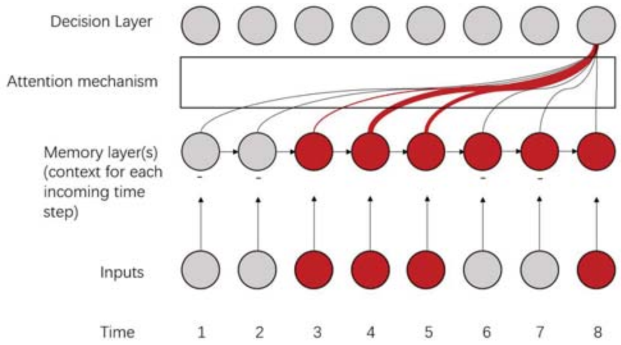


Fig. 5: Attention mechanism

Lin and his colleagues proposed a new model for extracting an interpretable sentence embedding by introducing self-attention and got an impressive result on the sentiment classification problem [23]. Considering the similarity between sentiment classification and HAR, we use this self-attention method in HAR to give the model the ability to generate weight distributions over the history of samples.

In this self-attention mechanism, there is no extra information and the model is only given one single sequence as input. Therefore the query, key and value for this self-attention mechanism are the same one, which is the output vectors of the LSTM layer.

Let H be a matrix consisting of output vectors $[h_1, h_2, \cdots, h_n]$ that the LSTM layer produced, where

n denotes the length. The representation γ of the input is formed by a weighted sum of these output vectors:

$$M = \tanh(H) \tag{11}$$

$$\alpha = \mathrm{softmax}(w^T M) \tag{12}$$

$$\gamma = H\alpha^T \tag{13}$$

where $H \in \mathbb{R}^{m \times n}$, m is the dimension of the hidden state, w is a trained parameter and w^T is the transpose of w. Softmax function takes $w^T M$ as input, and normalizes it into a probability distribution consisting of n probabilities. The dimensions of w, α, γ are m, n, m respectively.

We obtain the final temporal representation used for classification from:

$$h^* = \tanh(r) \tag{14}$$

3.4 Classifying

In this subsection, a softmax classifier is used to predict label $\hat{A}$ from a discrete set of classes A for a temporal sample S. The softmax layer takes the transformed h^* as input:

$$\hat{p}(y|S) = \mathrm{softmax}(W^{(S)} h^* + b^{(S)}) \tag{15}$$

$$\hat{y} = \arg\max_{y} \hat{p}(y|S) \tag{16}$$

where $W^{(S)}$ and $b^{(S)}$ denote the weighted matrix and the bias of the softmax layer respectively.

We choose the cross-entropy error function as cost function:

$$J(\theta) = -\frac{1}{d}\sum_{i=1}^{d} y_i \log(p_i) + \lambda \left\|\theta\right\|_2^2 \tag{17}$$

where $y \in \mathbb{R}^d$ is the one-hot represented ground truth in which all the elements are 0 except true label has 1 as its value, $p \in \mathbb{R}^d$ is the predicted probability that the observed sample belongs to class i (d is the number of target classes), θ is the parameters of the network, and the latter term of the loss function is an L_2 regularization.

3.5 Tricks for Optimization

Dropout, proposed by Hinton, is used to improve overfit on neural networks by ignoring units during the training phase of a certain set of neurons [24]. We apply dropout on the LSTM layer and attention layer. Also, the L_2 norm of the weights for weight decay is added in the loss function.

To prevent a sudden leveling off in the accuracy during learning, gradient clipping is employed with a certain gradient step norm [25]. Training details are further introduced in Section 4.2.

4 Experiments

In this section, we first describe the Opportunity data set. Then, we introduce the parameter settings of attention-based LSTM and evaluation metrics. Finally, we compare the proposed model against several different baseline methods and analyze the confusion matrix of the experiment. The computer for testing has an NVIDIA TITAN X (Pascal) which has 12 GB RAM and 3584 CUDA cores.

4.1 Opportunity Data Set

The Opportunity data set for HAR from wearable, object, and ambient sensors is a data set devised to benchmark human activity recognition algorithms. The data set contains activities from 4 subjects and each has 6 different runs. Five of them, termed activity of daily living, are composed by temporally unfolding situations in which a large number of action primitives occur. The remaining one, a drill run, is designed to generate a large number of scripted sequence instances. Notably, we use 17 mid-level gesture classes for predictions. This group contains the "NULL" class, which is common, for a total of 18 classes. Meanwhile, we remove useless features for gesture recognition and keep 113 features from the 242 features provided by the data set. To accelerate convergence, we use mean and variance normalization on the z-score with a standard deviation of 0.5. The normalization function is defined as follows:

$$x^* = \frac{x - \mu}{\sigma} \tag{18}$$

where μ and σ denote mean value and standard deviation, respectively.

After the operation of sliding window, there are a total of 18794 annotated samples, including 15497 time sequences for training, and 3297 for testing.

4.2 Parameter Settings and Evaluation Metrics

To determine the window length n and the dimension of hidden states m, we conduct a grid search over $n \in \{24, 48, 72, 96, 192\}$, $m \in \{16, 32, 64, 128, 256\}$. The combination of $n = 72, m = 128$ achieves the best performance over test data. Our model is trained using RMSProp [26] with a learning rate of 0.001 and decayed after every epoch, an L_2 regularization weight of 0.001, and a minibatch size 80. To prevent the gradients from getting too large, we employ gradient clipping with a maximal gradient step norm of 5. We evaluate the effect of dropout LSTM layer and dropout the attention layer, the model achieves a better performance, when the dropout rate is set as 0.5 and 0.7 respectively. The other parameters are initialized by sampling from a uniform distribution $U(-\epsilon, \epsilon)$. Pytorch is used for implementing our neural network models [27].

Due to the high imbalance of the data set, we use a weighted F1-score to evaluate results. The F1-score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. It can be defined as follows:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{19}$$

For this multi-class problem, we calculate metrics for each label, and find their average weighted by support (the number of true instances for each label). So, the F1-score is described as follows:

$$F_1 = 2 \times \sum_{i=1}^{d} \frac{N_i}{N_{total}} \frac{precision \times recall}{precision + recall} \tag{20}$$

where N_i is the true sample count of class i, d is the number of classes, and N_{total} is the total sample count of the data set.

Table 1: Confusion matrix for Opportunity data set using Att-LSTM

		Predicted labels																
		Other	Open Door 1	Open Door 2	Close Door 1	Close Door 2	Open Fridge	Close Fridge	Open Dishwasher	Close Dishwasher	Open Drawer 1	Close Drawer 1	Open Drawer 2	Close Drawer 2	Open Drawer 3	Close Drawer 3	Clean Table	Drink from Cup	Toggle Switch
True labels	Other	**2520**	15	2	4	4	13	9	8	7	2	9	3	8	2	7	8	35	5
	Open Door 1	1	**14**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	Open Door 2	1	0	**26**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Close Door 1	1	6	0	**9**	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	Close Door 2	1	0	2	0	**21**	0	0	0	0	0	0	0	0	0	0	0	0	0
	Open Fridge	17	0	0	0	0	**55**	1	0	0	0	0	0	0	0	0	0	1	0
	Close Fridge	6	0	0	0	0	2	**44**	0	0	0	0	0	0	0	0	0	0	0
	Open Dishwasher	15	0	0	0	0	1	0	**17**	0	0	0	1	0	0	0	0	0	0
	Close Dishwasher	11	0	0	0	0	0	0	0	**11**	2	0	0	1	0	0	0	0	0
	Open Drawer 1	3	0	0	0	0	0	0	0	0	**5**	3	1	2	0	0	0	0	0
	Close Drawer 1	1	0	0	0	0	0	0	0	0	2	**12**	0	0	0	0	0	0	0
	Open Drawer 2	3	0	0	0	0	1	0	0	0	3	1	**5**	1	1	0	0	0	0
	Close Drawer 2	1	0	0	0	0	0	0	0	0	1	0	2	**5**	0	0	0	0	0
	Open Drawer 3	1	0	0	0	0	0	0	0	0	1	0	8	1	**11**	1	0	0	0
	Close Drawer 3	1	0	0	0	0	0	0	0	0	0	0	0	9	3	**9**	0	0	0
	Clean Table	13	0	0	0	0	2	0	0	0	0	0	1	0	0	0	**16**	1	0
	Drink from Cup	30	0	0	0	0	2	1	1	0	0	0	0	0	0	0	0	**71**	0
	Toggle Switch	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**18**

4.3 Experimental Results

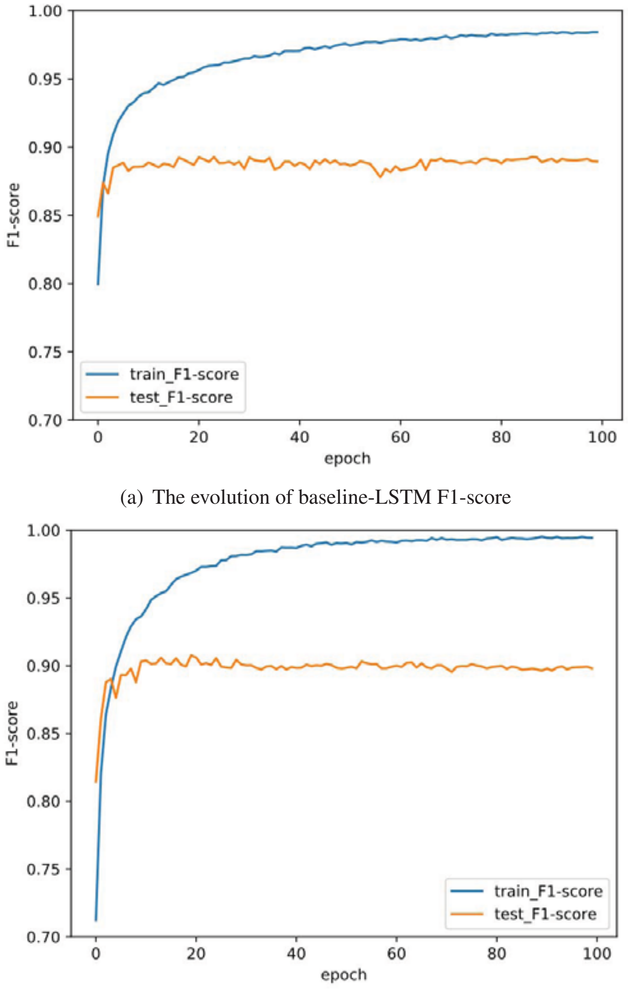
Table 2: F1-score of each algorithm

	Algorithms	F1-score
Opportunity Challenge Submission[10]	LDA	69%
	QDA	53%
	NCC	51%
	1NN	87%
	3NN	85%
	UP	64%
	NStar	84%
	SStar	86%
	CStar	88%
Deep Model	CNN[28]	85.1%
	BaselineLSTM	88.3%
	Bidir-LSTM[29]	89.4%
	Deep-Res-Bidir-LSTM[29]	90.2%
	Att-LSTM	**90.9%**

Table 1 compares our attention-based LSTM with other previously reported methods on the Opportunity data set. Generally speaking, LSTM models outperform others in the experiment. Our proposed attention-based LSTM (Att-LSTM) model yields an F1-score of 90.9%. When compared to the best submissions of the Opportunity challenge, our model improves the performance by 2.9%. Also, Att-LSTM model improves by 5.8% over the result reported by Yang using CNN [28]. Compared with the result of Deep-Res-Bidir-LSTM, which used residual architecture, it improves by 0.7% [29].

Table 2 shows the confusion matrix of the attention-based model with test data. The confusion matrix contains information about the true label and predicted label done by the model, to identify the nature of the classification errors, as well as their quantities. Each cell in the confusion matrix stands for the number of times that the gesture in the row is labeled as the gesture in the column. It can be seen that most misclassifications are related to the "other" category, as "other" class accounts for 75% of the data set. As for the other categories, our model achieves very good results, and only very few samples are classified as wrong labels.

Fig.6 shows the F1-score trend with the training data and testing data for baseline LSTM and Att-LSTM. It can be seen that both models eventually converge and there is no overfitting. The final F1-score of Att-LSTM is significantly higher than the baseline LSTM, and the amplitude is smaller. Overall, Att-LSTM shows better performance during the training process.



(a) The evolution of baseline-LSTM F1-score

(b) The evolution of att-LSTM F1-score

Fig. 6: The evolution of F1-score of Baseline-LSTM and Att-LSTM

5 Conclusion

In this paper, we propose an Att-LSTM network for HAR. The key idea of our approach is to allocate attention and learn inner relationship, by adjusting the weights that the model assigns to various inputs. Our proposed model can concentrate on the relevant temporal context so that it is more competitive for behavior sets with individually varying durations. In our experiments, the proposed model obtains superior performance over baseline models on Opportunity data set.

Future work will explore which parts of the attention layer are more decisive on classification. The effect of one-dimensional time-based convolution at one or some points in the LSTM cells will also be analyzed. CNNs show outstanding ability in exploring high-level features, which might further improve the performance of Att-LSTM. Finally, to verify whether Att-LSTM has a good generalization, we will apply our model to other data sets and time sequences classification problems.

References

[1] H. Kalantarian, C. Sideris, B. Mortazavi, Dynamic computation offloading for low-power wearable health monitoring systems, *IEEE Transactions on Biomedical Engineering*, 64(3): 621–628, 2017.

[2] S.H. Ahmed, D. Kim, Named data networking-based smart home, *Ict Express*, 2(3): 130–134, 2016.

[3] S.S. Rautaray, A. Agrawal, Vision based hand gesture recognition for human computer interaction: a survey, *Artificial Intelligence Review*, 43(1): 1–54, 2015.

[4] F.J. Ordóñez, D. Roggen, Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition, *Sensors*, 16(1): 115, 2016.

[5] D. Cook, KD. Feuz, NC. Krishnan, Transfer learning for activity recognition: A survey, *Knowledge and Information Systems*, 36(3): 537–556, 2013.

[6] L. Wang, Recognition of human activities using continuous autoencoders with wearable sensors, *Sensors*, 16(2): 189, 2016.

[7] O.D. Lara, MA. Labrador, A survey on human activity recognition using wearable sensors, *IEEE Communications Surveys and Tutorials*, 15(3): 1192–1209, 2013.

[8] Y. Bengio, Deep learning of representations: Looking forward, in *International Conference on Statistical Language and Speech Processing*, 1–37, 2013.

[9] A. Bulling, U. Blanke, B. Schiele, A tutorial on human activity recognition using body-worn inertial sensors, *ACM Computing Surveys (CSUR)*, 46(3): 33, 2014.

[10] R. Chavarriaga, H. Sagha, A. Calatroni, The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition, *Pattern Recognition Letters*, 34(15): 2033–2042, 2013.

[11] D. Anguita, A. Ghio, L. Oneto, A public domain dataset for human activity recognition using smartphones, *ESANN*, 2013.

[12] J. Wang, Y. Chen, S. Hao, Deep learning for sensor-based activity recognition: A survey, *Pattern Recognition Letters*, 2018

[13] J. Mantyjarvi, J. Himberg, T. Seppanen, Recognizing human motion with multiple acceleration sensors, in *2001 IEEE International Conference on Systems, Man and Cybernetics*, 2: 747–752, 2001.

[14] D.O. Olguın, AS. Pentland, J. Fahrenberg, Human activity recognition: Accuracy across common locations for wearable sensors, *Proceedings of 2006 10th IEEE International Symposium on Wearable Computers, Montreux, Switzerland*, 11–14, 2006.

[15] J.R. Kwapisz, GM. Weiss, SA. Moore, Activity recognition using cell phone accelerometers, *ACM SigKDD Explorations Newsletter*, 12(2): 74–82, 2011.

[16] C.A. Ronao, SB. Cho, Human activity recognition with smartphone sensors using deep learning neural networks, *Expert Systems with Applications*, 59: 235–244, 2016.

[17] N.Y. Hammerla, S. Halloran, T. Ploetz, Convolutional, and recurrent models for human activity recognition using wearables, *arXiv preprint arXiv:1604.08880*, 2016.

[18] O. Banos, JM. Galvez, M. Damas, Window size impact in human activity recognition, *Sensors*, 14(4): 6474–6499, 2014.

[19] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Computation*, 9(8): 1735–1780, 1997.

[20] J.K. Chorowski, D. Bahdanau, D. Serdyuk, Attention-based models for speech recognition, *Advances in Neural Information Processing Systems*, 577–585, 2015.

[21] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, *arXiv preprint arXiv:1409.0473*, 2014.

[22] K.M. Hermann, T. Kocisky, E. Grefenstette, Teaching machines to read and comprehend, *Advances in Neural Information Processing Systems*, 1693–1701, 2015.

[23] Z. Lin, M. Feng, CN. Santos, A structured self-attentive sentence embedding, *arXiv preprint arXiv:1703.03130*, 2017.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, Dropout: a simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research*, 15(1): 1929–1958, 2014.

[25] R. Pascanu, T. Mikolov, Y. Bengio, Understanding the exploding gradient problem, *CoRR, abs/1211.5063*, 2012.

[26] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, *COURSERA: Neural Networks for Machine Learning*, 4(2): 26–31, 2012

[27] A. Paszke, S. Gross, S. Chintala, Automatic differentiation in pytorch, 2017.

[28] J. Yang, MN. Nguyen, PP. San, Deep convolutional neural networks on multichannel time series for human activity recognition, *Ijcai*, 4(2): 15: 3995–4001, 2015.

[29] Y. Zhao, R. Yang, G. Chevalier, Deep residual bidir-LSTM for human activity recognition using wearable sensors, *arXiv preprint arXiv:1708.08989*, 2017.