Compare of Machine Learning And Deep Learning Approaches for Human Activity Recognition

Babak Moradi, Mohammad Aghapour and Afshin Shirbandi

September 12, 2019

Compare of Machine Learning And Deep Learning Approaches for Human Activity Recognition

Babak Moradi
Department of Computer (Computer engineering), Kermanshah Branch, Islamic Azad University, Kermanshah, Iran
Email: babak.moradi1988@gmail.com

Mohammad Aghapour
Department of Electrical, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
E-mail: M_ghapour@aut.ac.ir

Afshin Shirbandi
Robotic Research Center, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
E-mail: Afshin_shirbandi@aut.ac.ir

Abstract

Nowadays analyze of human activity and human behavior can be useful for software design especially for patients. So, human activity recognition is important. The aim of this research was find the best algorithm for human activity recognition. We used Logistic Regression, SVM with RBF kernel; CNN, LSTM, Bi-Directional LSTM and CNN-LSTM algorithms for analyze the data. In the data analyze the accuracy and training time measured and compared. The most accuracy belonged to the CNN-LSTM and Bi-Directional LSTM and the least training time belonged to the SVM with RBF kernel.

Keywords: Human activity recognition, Logistic Regression, SVM with RBF kernel, CNN, LSTM, Bi-Directional LSTM, CNN-LSTM

1. Introduction

Human activity recognition (HAR) is a classification task that makes use of time-series data from devices such as accelerometers and gyroscopes, preprocess these signals, extract relevant and discriminative features from them, and finally, recognize activities by using a classifier. Especially those gathered from sensors, time-series data have a strong 1D structure in that they are very highly correlated to temporally nearby local readings [1, 2].

Human activity recognition (HAR) is based on the assumption that specific body movements translate into characteristic sensor signal patterns, which can be sensed and classified using machine learning techniques. In this article, we are interested in wearable (on-body) sensing, as this allows activity and context recognition regardless of the location of the user [3].

The recognition of human activities has been approached in two different ways, namely using external and wearable sensors. In the former, the devices are fixed in predetermined points of interest, so the inference of activities entirely depends on the voluntary interaction of the users with the sensors. In the latter, the devices are attached to the user [4].

Smart homes [5-9] are a typical example of external sensing. These systems can recognize fairly complex activities (e.g., eating, taking a shower, washing dishes, etc.), because they rely on data from a number of sensors placed in target objects which people are supposed to interact with (e.g., stove, faucet, washing machine, etc.). Nonetheless, nothing can be done if the user is out of the reach of the sensors or they perform activities that do not require interaction with them. Additionally, the installation and maintenance of the sensors usually entail high costs [4].

Cameras have also been employed as external sensors for HAR. In fact, the recognition of activities and gestures from video sequences has been the focus of extensive research [10-13]. This is especially suitable for security (e.g, intrusion detection) and interactive applications. A remarkable example, and also commercially available, is the Kinect game console [14] developed by Microsoft. It allows the user to interact with the game utilizing gestures, without any controller device [4].

Recently, deep learning has emerged as a family of learning models that aim to model high-level abstractions in data [15, 16]. In deep learning, a deep architecture with multiple layers is built up for automating feature design. Specifically, each layer in deep architecture performs a nonlinear transformation on the outputs of the previous layer, so that through the deep learning models the data are represented by a hierarchy of features from low-level to high-level [17].

We decided to compare several algorithms for HAR. So we selected Logistic Regression, SVM with RBF kernel, CNN, LSTM, Bi-Directional LSTM, and CNN-LSTM.

In this paper, in section 2, we explain about methods and algorithms that we used for this paper. In section 3, we report the results of methods and algorithms and compare them, and in section 4, we discuss results.

2. Methods

hyperparameter optimization

hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. hyperparameter optimization finds a tuple of hyperparameters that yields an optimal model which minimizes a predefined loss function on given independent data. The objective function takes a tuple of hyperparameters and returns the associated loss.in this work we use this approach in all algorithm find best model.

2. 1. Logistic Regression

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous. Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval, or ratio-level independent variables [18]. Table (1) shows Logistic regression models that find in hyperparameter tuning.

Table (1). Logistic regression models find by hyperparameter tuning approach.

Model Name	penalty	C
$\mathbf{M_0}$	L2	0.01
$\mathbf{M_1}$	L1	0.01
$\mathbf{M_2}$	L2	0.2
$\mathbf{M_3}$	L1	0.2
$\mathbf{M_4}$	L2	1
$\mathbf{M_5}$	L1	1
$\mathbf{M_6}$	L2	5
$\mathbf{M_7}$	L1	5
$\mathbf{M_8}$	L2	10
$\mathbf{M_9}$	L1	10
$\mathbf{M_{10}}$	L2	15
$\mathbf{M_{11}}$	L1	15

2. 2. SVM with RBF kernel

RBF kernel function is a universal kernel function, and it can be applied to any of the distribution of the samples through the choice of parameters. It has been more and more used in the nonlinear mapping of the support vector machine.

RBF kernel function expression is:

$K(x_i, x_j) = exp(-\gamma\|x_i - x_j\|)^2$, and the corresponding minimization problem of support vector machine (SVM) is:

$$\min_{a_i} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j a_i a_j exp(-\gamma\|x_i - x_j\|)^2 - \sum_{i=1}^{n} a_i \qquad (1) \quad [19]$$

Among s.t. $\sum_{i=1}^{n} y_i a_o = 0, \;\; 0 \leq a_i \leq C$ [19]

So the minimum value of the type depends on the choice of parameters(C,γ). So the best parameters which we choose like this can make the classifier performance of SVM is the best, that is, its great promotion is most, and it is the promoting error rate is lowest.

The function of C is adjusting the confidence interval range of the learning machine in the specified data subspace, and the optimization of C is different in the different data subspace. The change of kernel parameters γ is actually changing the mapping function implicitly, and then changing the complexity level of the distribution of the sample data subspace distribution, which is the biggest VC dimension of linear classification face, it also decided the linear classification to minimum error. The research of Vapnike and others shows that the kernel parameter and the error punish factor are the key factor of influencing SVM performance [19]. Table (2) shows SVM models that find in hyperparameter tuning.

Table (2).SVM models find by hyperparameter tuning approach.

Model Name	gamma	C
$\mathbf{M_0}$	0.0725	2
$\mathbf{M_1}$	0.0725	8
$\mathbf{M_2}$	0.0725	16
$\mathbf{M_3}$	0.156	2
$\mathbf{M_4}$	0.156	8
$\mathbf{M_5}$	0.156	16
$\mathbf{M_6}$	0.92	2
$\mathbf{M_7}$	0.92	8
$\mathbf{M_8}$	0.92	16

2. 3. CNN

CNN's aim to introduce a degree of locality in the patterns matched in the input data and to enable translational invariance for the precise location (i.e., time of occurrence) of each pattern within a frame of movement data. We explore the performance of convolutional networks and follow suggestions by [20] in architecture and regularization techniques. Each CNN contains at least one temporal convolution layer, one pooling layer, and at least one fully connected layer before a top-level softmax-group. The temporal convolution layer corresponds to a convolution of the input sequence with different kernels (feature maps) of width. Subsequent max-pooling is looking for the maximum within a region of width and corresponds to a subsampling, introducing translational invariance to the system [21].

In this work, we use a CNN architecture with two convolutional layers and two dropout layer to avoiding overfitting. The first drop out layer has put between convolutional layers and second drop out layer has put before the dense layer.

2. 4. LSTM

LSTM network models are a type of recurrent neural network that is able to learn and remember over long sequences of input data. They are intended for use with data that is comprised of long sequences of data, up to 200 to 400-time steps. They may be a good fit for this problem.

The model can support multiple parallel sequences of input data, such as each axis of the accelerometer and gyroscope data. The model learns to extract features from sequences of observations and how to map the internal features to different activity types.

The benefit of using LSTMs for sequence classification is that they can learn from the raw time series data directly, and in turn do not require domain expertise to engineer input features manually. The model can learn an internal representation of the time series data and ideally achieve comparable performance to models fit on a version of the dataset with engineered features.

2. 5. Bi-Directional LSTM

Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems.

In problems where all time steps of the input sequence are available, Bidirectional LSTMs train two instead of one LSTMs on the input sequence. The first on the input sequence as-is and the second on a reversed copy of the input sequence. This can provide additional context to the network and result in faster and even fuller learning on the problem [22].

2. 6. CNN-LSTM

The CNN-LSTM architecture involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to support sequence prediction.

CNN-LSTMs were developed for visual time series prediction problems and the application of generating textual descriptions from sequences of images.

The CNN LSTM model will read subsequences of the main sequence in as blocks, extract features from each block, then allow the LSTM to interpret the features extracted from each block.

2.7. Dataset

We used Human Activity Recognition Using Smartphones Data Set. In this dataset, the experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers were selected for generating the training data and 30% the test data. Number of Instances was 10299 [23].

3. Results

For comparing all the algorithms, we decided to select accuracy and time of training. Because of this two feature, we can have a better selection.

After using all six algorithms on the dataset, results were as that showed in figure (1) parts a to f.

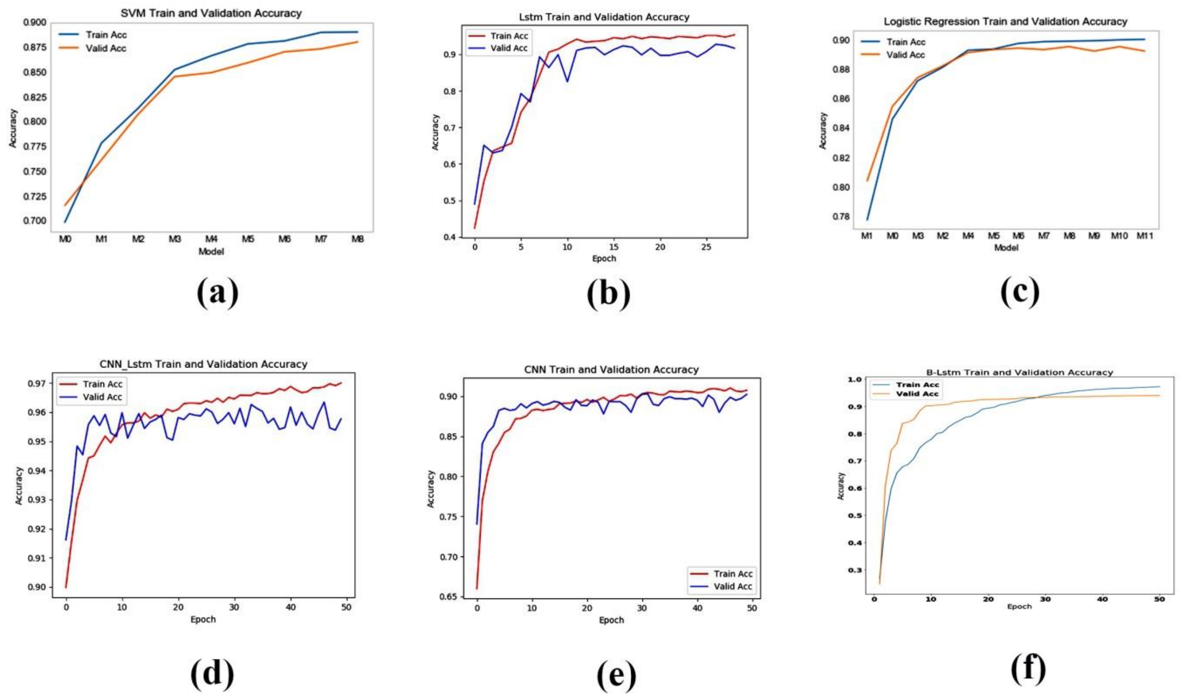

Figure (1): a) Accuracy of SVM with RBF kernel algorithm b) Accuracy of LSTM algorithm c) Accuracy of Logistic Regression algorithm d) Accuracy of CNN-LSTM algorithm e) Accuracy of CNN algorithm f) Accuracy of Bi-Directional LSTM algorithm.

In the table (1) we showed results of accuracy of each algorithm.

Table 1: accuracy of algorithms that used for Human activity recognition

Algorithm	LSTM	CNN	CNN-LSTM	Bi-Directional LSTM	SVM with RBF kernel	Logistic Regression
Accuracy	95	91	97	97	89	90

In figure (2) and table (2), we showed results of the time of training for each algorithm.

Table 2: Time of training of algorithms that used for Human activity recognition

Algorithm	LSTM	CNN	CNN-LSTM	Bi-Directional LSTM	SVM with RBF kernel	Logistic Regression
Accuracy	322	245	957	575	78	93

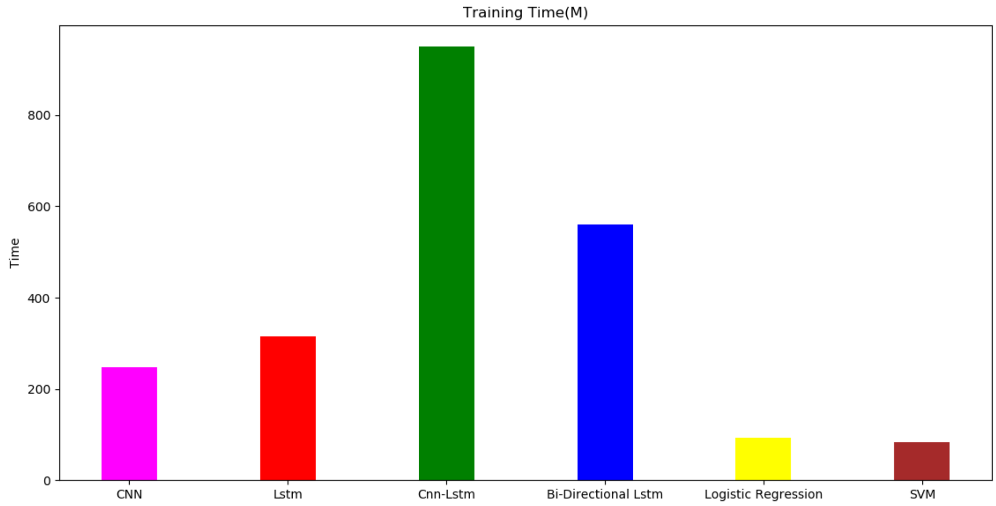

Figure 2: training Time of algorithms for Human activity recognition

4. Conclusion

In this work we used Logistic Regression, SVM with RBF kernel, CNN, LSTM, Bi-Directional LSTM and CNN-LSTM Methods and compared them with together. In accuracy feature CNN-LSTM algorithm and Bi-Directional LSTM algorithm had most result. In time of training SVM with RBF kernel algorithm had least time for training and CNN-LSTM had most time for training. When we need more accuracy, we can use CNN-LSTM algorithm or Bi-Directional LSTM algorithm, and when time is important to us, we can use SVM with RBF kernel algorithm.when important of accuracy and time is same for us,The best choice is LSTM.

References

[1] C. A. Ronao and S.-B. Cho, "Deep convolutional neural networks for human activity recognition with smartphone sensors," in *International Conference on Neural Information Processing*, 2015, pp. 46-53: Springer.

[2] Y. LeCun, Y. J. T. h. o. b. t. Bengio, and n. Networks, "Convolutional networks for images, speech, and time series," vol. 3361, no. 10, p. 1995, 1995.

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

[4] O. D. Lara, M. A. J. I. c. s. Labrador, and tutorials, "A survey on human activity recognition using wearable sensors," vol. 15, no. 3, pp. 1192-1209, 2012.

[5] T. Kasteren, G. Englebienne, B. J. P. Kröse, and u. Computing, "An activity monitoring system for elderly care using generative and discriminative models," vol. 14, no. 6, pp. 489-498, 2010.

[6] A. Tolstikov *et al.*, "Comparison of fusion methods based on dst and dbn in human activity recognition," vol. 9, no. 1, pp. 18-27, 2011.

[7] J. Yang, J. Lee, J. J. J. o. C. S. Choi, and Technology, "Activity recognition based on RFID object usage for smart mobile devices," vol. 26, no. 2, pp. 239-246, 2011.

[8] J. Sarkar, Y.-K. Lee, and S. J. A. I. Lee, "GPARS: A general-purpose activity recognition system," vol. 35, no. 2, pp. 242-259, 2011.

[9] J. Hong and T. Ohtsuki, "A state classification method based on space-time signal processing using SVM for wireless monitoring systems," in *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, 2011, pp. 2229-2233: IEEE.

[10] P. Turaga, R. Chellappa, V. S. Subrahmanian, O. J. I. T. o. C. Udrea, and S. f. V. technology, "Machine recognition of human activities: A survey," vol. 18, no. 11, p. 1473, 2008.

[11] J. Candamo, M. Shreve, D. B. Goldgof, D. B. Sapper, and R. J. I. t. o. i. t. s. Kasturi, "Understanding transit scenes: A survey on human behavior-recognition algorithms," vol. 11, no. 1, pp. 206-224, 2009.

[12] C. Joseph, S. Kokulakumaran, K. Srijeyanthan, A. Thusyanthan, C. Gunasekara, and C. Gamage, "A framework for whole-body gesture recognition from video feeds," in *2010 5th International Conference on Industrial and Information Systems*, 2010, pp. 430-435: IEEE.

[13] M. A. R. Ahad, J. Tan, H. Kim, and S. Ishikawa, "Human activity recognition: Various paradigms," in *2008 International Conference on Control, Automation and Systems*, 2008, pp. 1896-1901: IEEE.

[14] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," in *CVPR 2011*, 2011, pp. 1297-1304: Ieee.

[15] Y. J. F. Bengio and t. i. M. Learning, "Learning deep architectures for AI," vol. 2, no. 1, pp. 1-127, 2009.

[16] L. J. A. T. o. S. Deng and I. Processing, "A tutorial survey of architectures, algorithms, and applications for deep learning," vol. 3, 2014.

[17] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[18] W. Liu, H. Liu, D. Tao, Y. Wang, and K. J. S. P. Lu, "Multiview Hessian regularized logistic regression for action recognition," vol. 110, pp. 101-107, 2015.

[19] S. Han, C. Qubo, and H. Meng, "Parameter selection in SVM with RBF kernel function," in *World Automation Congress 2012*, 2012, pp. 1-4: IEEE.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. J. T. j. o. m. l. r. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," vol. 15, no. 1, pp. 1929-1958, 2014.

[21] N. Y. Hammerla, S. Halloran, and T. J. a. p. a. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," 2016.

[22] Z. Yu *et al.*, "Using bidirectional lstm recurrent neural networks to learn high-level abstractions of sequential features for automated scoring of non-native spontaneous speech," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 338-345: IEEE.

[23] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International workshop on ambient assisted living*, 2012, pp. 216-223: Springer.