Human Activity Recognition using UniMiB Data Shar

Sheeza Batool,Laiba Shoukat
Department of Computer Science
University of the Punjab
Lahore, Pakistan
mscsf21m511@pucit.edu.pk,mscsf21m513@pucit.edu.pk

Abstract

Getting a good feature representation of data is paramount for Human Activity Recognition (HAR) using wearable sensors. An increasing number of feature learning approaches-in particular deep-learning based—have been proposed to extract an effective feature representation by analyzing large amounts of data.We implemented the codes and implementation details to make both the reproduction of the results reported in this paper. Our work on UniMiB-SHAR dataset highlight the effectiveness of deep-learning architectures involving Multi-Layer-Perceptron (MLP), Convolutional Neural Network (CNN) to obtain features characterising in the data.

1 Introduction

Human Activity Recognition (HAR) is a research topic which has attracted an increasing amount of attention from the research community, in the wake of the development and spread of increasingly powerful and affordable mobile devices or wearable sensors. The main goal of HAR is automatic detection and recognition of activities from the analysis of data acquired by sensors. HAR finds potential applications in numerous areas, ranging from surveillance tasks for security, to assistive living and improvement of the quality of life or gaming.[1].

1.1 UnimiB-SHAR:

The UniMiB-SHAR dataset (University of Milano Bicocca Smartphone-based HAR) aggregates data from 30 subjects (6 male and 24 female) acquired using the 3D accelerometer of a Samsung Galaxy Nexus I9250 smartphone (S = 3). The data are sampled at a frequency of 50 Hz, and split in 17 different classes, comprising 8ADLs and 7 "falling" actions as shown in Table 1. Each activity is either performed 2 or 6 times, with half of them having the subject place the smartphone in his/her left pocket, and the other half in his/her right pocket.

1.2 Multi-Layer Perceptron:

Multi-Layer-Perceptron (MLP) is the simplest class of ANN, and involves a hierarchical organisation of neurons in layers. MLPs comprise at least three fully-connected layers (also called dense layers) including an input, one or more intermediate (hidden) and an output layer, as shown in Figure 1. Each neuron of a fully-connected layer takes the outputs of all neurons of the previous layer as its inputs. Considering that the output values of hidden neurons represent a set of features extracted from the input of the layer they belong to, stacking layers can be seen as extracting features of an increasingly higher level of abstraction, with the neurons of the nth layer outputting features computed using the ones from the (n 1)th layer.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

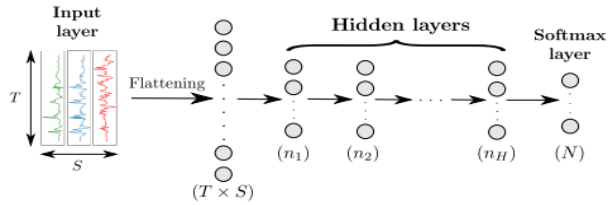

Figure 1: Architecture of a MLP model with H hidden layers for sensor-based HAR

Table 1: Classes of the UniMiB-SHAR dataset and their respective proportions. ADLs and falls are provided in the left and right column, respectivel

Class	Proportion	Class	Proportion
Standing Up from Sitting	1.30	Falling Forward	4.49
Standing Up from Laying	1.83	Falling Left	4.54
Walking	14.77	Falling Right	4.34
Running	16.86	Falling Backward	4.47
Going Up	7.82	Falling Backward Sitting-chair	3.69
Jumping	6.34	Falling with Protection Strategies	4.11
Going Down	11.25	Falling and Hitting Obstacle	5.62
Lying Down from Standing	2.51	Syncope	4.36
Sitting Down	1.70		

1.3 Convolutional Neural Networks:

CNNs comprise convolutional layers featuring convolutional neurons. The kth layer is composed of nk neurons, each of which computes a convolutional map by sliding a convolutional kernel (f(k)T, f(k)S) over the input of the layer (indicated in red in Figure 2). Convolutional layers are usually used in combination with activation layers, as well as pooling layers. The neurons of the latter apply a pooling function (e.g., maximum, average, etc.) operating on a patch of size (p(k)T, p(k)S) of the input map to downsample it (indicated in blue in Figure 2, to make the features outputted by neurons more robust to variations in temporal positions of the input data. Convolution and pooling operations can either be performed on each sensor channel independently (f(k)S = 1 and/or p(k)S = 1) or across all sensor channels (f(k)S = S and/or p(k)S = S).Similarly to regular dense layers, convolutional-activation-pooling blocks can be stacked to craft high-level convolutional features. Neurons of stacked convolutional layers operate a convolutional product across all the convolutional maps of the previous layer. For classification purposes, a fully-connected layer after the last block can be added to perform a fusion of the information extracted from all sensor channels. The class probabilities are outputted by a softmax layer appended to the end of the network.

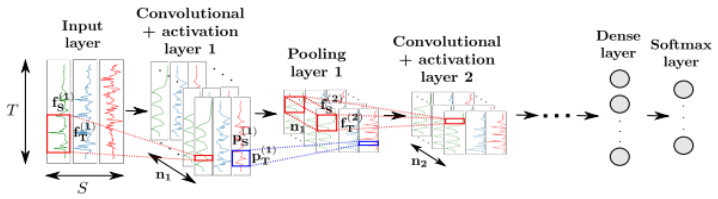

Figure 2: Architecture of a CNN model for sensor-based HAR.

2 Related Work

Traditionally, the fields of smart home technology and activity recognition are highly overlapping. Smart homes provide the hardware to collect and process relevant data, while activity recognition approaches extract semantic meanings from the collected data by adopting machine learning methods. Our system is not an exception to this overlap. For this reason, we want to point out highly relevant work from the smart home area and research approaches from the field of activity recognition. We read the following papers to have an extended level of understanding of background knowledge and techniques used for the recognition of human activity recognition. [2] [3] [4]

3 Methodology

In this project,we are using the feature encoding technique, any other researcher have not used the feature encoding technique to extract the main features from the data set. They diectly applied CNN, RNN, and Neural network.Before applying the super vector feature encoding technique,generate codebooks for all sensor data in all dimensions. Codebooks are generated using local features, which get through the sliding window technique, l is sliding window size and s stride.

3.1 Feature Extraction/ Local Descriptors:

Low-level local features have become popular in action recognition due to their robustness to background clutter and independence in detection and tracking techniques. Local features extract through the sliding window technique, the window size is 40, and the stride size is 10. Window slide on 11771 sample data through sliding we get the local feature of every row of every axis separately.

3.2 Codebook Generation:

Codebook Generation is implemented through GMM(Gaussian mixture models). A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. A mean that defines its center.

3.3 Feature Encoding:

The technique is using in this project is VLAD(Vector of Lo- cally Aggregated Descriptors), which are Supervector-based encoding methods. The VLAD algorithm can be regarded as a simplified FV. Its main method is to train a small codebook through a clustering method, subtracting the codebook from the nearest visual codeword of input data X.

3.4 Classification:

Support Vector Machines (SVC) with a radial basis kernel; For classification, we are using the SVM model to classify HAR. we are classifying Activities of Daily Living and Fall activities.

3.5 MLP:

We used a MLP with three hidden layers with Rectified Linear Units (RELU) activations, taking vectors obtained by flattening frames of data as inputs. We noticed that the addition of a batch normalization layer significantly improved the classification results. Batch normalization layers standardize the inputs of the subsequent layers by computing parameters on batches of training data to normalize each sensor channel independently.The final architecture used for our MLP models thus consists of a batch normalization layer, followed by three fully-connected layers with RELU activations, and an output layer with a softmax activation providing estimations of probabilities for each class.

3.6 CNN:

Our CNN architecture involves three consecutive blocks, each including a convolutional, RELU activation and max-pooling layers. Each convolutional kernel performs a 1D convolution over the time dimension, for each sensor channel independently. Similarly to MLP, adding a batch normalization layer right after the input layer yields significant performance improvements. The model selected in the end comprises in order a batch normalization layer, three blocks of convolutional-RELU-pooling layers, a fully-connected layer and a sigmoid and softmax layer for binary and Multiclass respectively.

4 Experiments and Results

4.1 Qualitative evaluation

The feature encoding technique Vlad applied resulted it better performance of features in case of binary class and a bit less in Multiclass due to different size of subset classes. The MLP layers for binary Classification is two layers with 50 neurons at each layer and for Multiclass classification three layers with 500 neurons at each layer are being used.The 1D convolution with 2 layers is being used in both multi and binary class classification. The filter size is 64 and kernel size is three with activation function relu.

4.2 Quantitative evaluation

Table 2: Learning Model's Accuracy

Class	Model	Training Accuracy	Testing Accuracy
Binary Class	SVM	82.8 percent	81.2 percent
Binary Class	MLP	85 percent	79.32 percent
Binary Class	CNN	99.54 percent	98.9 percent
Multiclass	SVM	54.9 percent	35 percent
Multiclass	MLP	70 percent	35 percent
Multiclass	CNN	98 percent	63.5 percent

5 Discussion

Between two deep learning models used, CNN performed better than MLP. By reading the literature we also seen other models performance which mean using better hybrid deep learning model could result in better performance.The feature encoding technique performed better at binary class but less better performance at multiclass.

6 Conclusion

In this report, we defined an evaluation framework which fixes all stages of the ARC including the feature extraction step to rigorously compare feature learning methods for sensor-based HAR. We applied this framework on the UniMiB-SHAR, to compare and evaluate the performances of two deep learning methods. The results indicate that a deep-learning architecture featuring convolutional Neural Network is flexible and robust approach, yielding top performances on the dataset. Changes in hyper-parameters related to the other steps of the ARC, such as the frame size or the number of sensor channels did not change the overall ranking of the feature learning models. The codebook approach also showed to be efficient for an energy-based segmentation case, such as the UniMiB-SHAR dataset. The fusion of features learned by both approaches further improved the results. All research materials used in our studies (i.e., datasets, codes and instructions on how to use them) are also provided to allow the reproduction of our results, and let other researchers test their own features on the datasets we used and rigorously compare their performances to ours.

References

[1] Liming Chen, Jesse Hoey, Chris D Nugent, Diane J Cook, and Zhiwen Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):790–808, 2012.

[2] Frédéric Li, Kimiaki Shirahama, Muhammad Adeel Nisar, Lukas Köping, and Marcin Grzegorzek. Comparison of feature learning methods for human activity recognition using wearable sensors. *Sensors*, 18(2):679, 2018.

[3] Lukas Köping, Kimiaki Shirahama, and Marcin Grzegorzek. A general framework for sensor-based human activity recognition. *Computers in biology and medicine*, 95:248–260, 2018.

[4] Daniela Micucci, Marco Mobilio, and Paolo Napoletano. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, 7(10):1101, 2017.