Long-term activity detection from the wrist

Kaushik Goud Chandapet, Nishanth Iruthayaraj, Venkata Sriram Pallav Pingala
University of Siegen
August 13, 2021

Abstract

Aiming at building and comparing various Classical and Deep neural network methods to classify or recognise basketball game related activities. At first, data is collected using three sensors for two different subjects, which is then pre-processed for the model. Based on this data, various classification methods such as KNN, SVM and deep learning methods such as CNN, LSTM are developed to recognise an unknown activity. Finally, a brief comparison of the models based on these results was carried out.

Keywords - Human Activity Recognition, Convolutional neural networks, Long short-term memory, KNN and SVM

1 Introduction

Human activity recognition is one of the most researched field in ubiquitous computing domain. Human activity is captured with various sensors. These sensors include accelerometers, gyroscopes and magnetometers etc., which are predominantly worn on body. The aim is to accurately and effectively classify the data into various activities and build a concurrent model for the activity recognition.

Here we have used 3 devices with embedded sensors and obtained the data from 2 participants. One E-sense device is attached to the left ear and two bangle.js watches are attached to right wrist and right ankle respectively. Data from these 3 sensors is not synchronous as they are running at different frequencies. Hence the data needs to be pre-processed before feeding it into the various machine learning models.

This paper describes the various machine learning/ deep learning techniques to effectively classify the Human activities. It also summarizes the effectiveness of each model and contrasts them with the other models in terms of metrics like precision, f1 score and accuracy.

2 Data pre-processing

Data Pre-processing is the step in which the data gets transformed, or Encoded, to bring it to such a state that now the machine can easily parse it.

2.1 Data Characteristics

The dataset consists of the sensor readings obtained from 3 mentioned devices for 2 participants. Each Bangle.js sensor provides accelerometer readings in 3 spatial dimensions X,Y,Z and the E-sense device provides accelerometer and gyroscope sensor readings each in 3 dimensions.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Device	Sensors	Frequency	Position	Activities Performed
eSense	Accelerometer, Gyroscope	50 Hz ±8g	On left ear	shooting, layups, dribbling, running, walking and null
Bangle.js Watch	Accelerometer	92 Hz ±8g	On right wrist and right ankle	shooting, layups, dribbling, running, walking and null

Figure : Data Characteristics Summary

2.2 Data Pre-processing Pipeline

The processing of the above data involved the following pipeline

1. The unix-timestamp of the E-sense device readings is decoded into Date-Time format of Local timezone to match with that of Bangle.js sensors.
2. The Timestamps of the Bangle.js device readings are corrected based on the sensor frequency.
3. As these devices are running at different frequencies and to match both E-sensor and Bangle.js sensors data, up-sampling is performed by means of interpolation to a combined frequency of 100 Hz.
4. Concatenated data from all the three device sensors into a single dataframe per subject.
5. The labels are then set(corrected) manually as per the timestamps from videos.
6. Data from both the subjects is concatenated into a single dataframe. The final dataset is Normalized using Zero mean and Unit Variance standardization.
7. Sliding window is applied on the final Dataset.

Sliding Window: To classify an activity we need the activity data for a period of time. Classification can't be done on a single sample of data but requires a collection of samples of short periods of time. This collection of samples of fixed small duration of time is called window or frame. The number of samples in that duration is termed as the window length or the frame size. The dataframe is divided into multiple windows and each window is affixed with one label specifying the activity performed in that window. The windows are created which are overlapping with a difference between each window starting sample, of fixed duration called the hop size. This can also be expressed in terms of overlapping ratio. This overlapping is done to get more windows given as an input to machine learning models for better classification during the training. For our experiment, we choose Window size: 100, Overlap-ratio: 70.

3 Model Description and Architectures

There are many different types of classical machine learning algorithms and Deep Neural networks that can be applied to the Human Activities Recognition problem. Some of the famous architectures that we have worked on are:

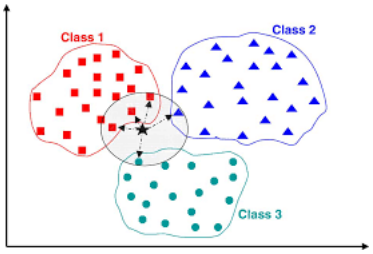
3.1 K Nearest Neighbors

K nearest neighbors is one of the simplest Machine Learning algorithms that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).

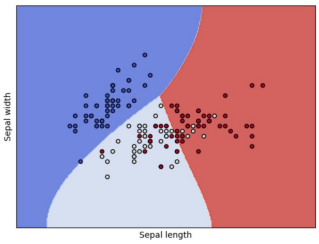
It works by finding the distances between a query and all the examples in the data, selecting the specified number of examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression). Thus, it assumes that similar things exist in close proximity.

Choosing the right K for our data is done by trying several Ks and picking the one that works best. For our experiment, we have selected k(n-neighbors) = 5.

This can be used to solve both classification and regression problems. It's easy to implement and understand, but has a major drawback of becoming significantly slower as the size of that data in use grows.



(a) K Nearest Neighbors



(b) Support vector machines

3.2 Support vector machines

Support vector machines is also a simple Machine Learning algorithm whose objective is to find a hyper-plane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

It works really well with a clear margin of separation and is effective in high dimensional spaces. It is effective in cases where the number of dimensions is greater than the number of samples.

It doesn't perform well when we have large data set because the required training time is higher or when the data set has more noise i.e. target classes are overlapping.

Similar to kNN, this can be used for both classification or regression challenges. For our experiment, we choose kernel='rbf', C=1, gamma='auto'.

3.3 Convolution Neural Network

Convolution Neural Network is a type of deep neural network which is used to analyze the visual data like images, this can also be used to analyse the time series data.

The input is an image or a frame , which consists of the data values of the pixels or the sensor values grouped together to preserve the time consistency. A filter/ kernel is used to slide along the input features and provide translation equivalent responses(convoluted outputs) known as feature maps.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 99, 11, 16)        80
_________________________________________________________________
dropout (Dropout)            (None, 99, 11, 16)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 98, 10, 32)        2080
_________________________________________________________________
dropout_1 (Dropout)          (None, 98, 10, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 31360)             0
_________________________________________________________________
dense (Dense)                (None, 128)               4014208
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_2 (Dense)              (None, 6)                 390
=================================================================
Total params: 4,025,014
Trainable params: 4,025,014
Non-trainable params: 0
```

Figure 2: Convolution Neural Network architecture

CNN have multiple layers; including convolutional layer, non-linearity layer, pooling layer and dense layer. The convolutional and fully-connected layers have parameters but pooling and non-linearity layers don't have parameters. For our experiment, this architecture consists of a deep neural network framework having sequential model with two conv2d layers followed by 3 dense layers with flatten and dropout layers in between. The Convolutional kernel of size (2,2) is added with Relu activation function.

3.4 Long short-term memory Neural Network

Long short-term memory (LSTM) is one of the variant of Recurrent neural network (RNN) published and is a well-suited network to bridge time lags of unknown size between different events. Since traditional RNN is susceptible to problems such as vanishing/exploding gradients, LSTM network has overcome such drawbacks with a new structure called memory cell as shown in fig a. It comprises four different components namely a cell state, a forget gate, an input gate and an output gate which interacts within the cell unit and generates an output for next memory cells. The recursive characteristics of the cell unit stores the information from previous states within the memory cell. Memory cells are updated by means of following equations at every time step t.

x_t is the input to the memory cell

W_f, W_i, W_C, W_o and b_f, b_i, b_C, b_o are the weight matrices and bias vectors of corresponding layers of a cell

Forget gate f_t manipulate the cell state to remember or forget its previous states at time t.

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f)$$

Input gate i_t decides values to be updated on the cell state with vector of new candidate $\tilde{\mathrm{C}}_t$ at time t

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_f$$

$$\tilde{\mathrm{C}}_t = \tanh(W_C * [h_{t-1}, x_t] + b_C)$$

The old cell state C_{t-1} is then updated into the new state C_t

$$C_t = f_t * C_{t-1} + i_t * \tilde{\mathrm{C}}_t$$

Output gate o_t value is computed which is combined with new state C_t to output the memory cells

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

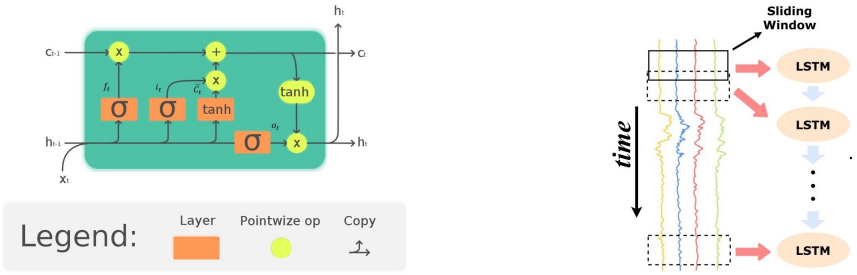


Figure 3: LSTM

The model architecture has first hidden layer which contains 26 LSTM cells for feature extraction from input data as shown in fig b followed by 0.1 percentage of dropout. The output of our hidden LSTM layer is flattened and fed into two fully connected layers. The first and second hidden layers of fully connected network contains 130 and 850 neurons with 0.2 and 0.3 percentage of dropout respectively. All the hidden layers are activated by 'relu' activation function except the output layer which acts as multi classifier by using 'softmax' activation function.

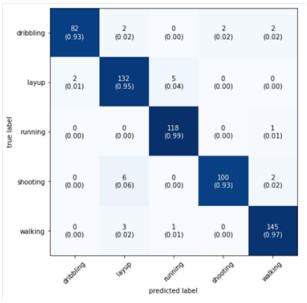
4 Result Analysis

The performance of different models on the given dataset is summarized in the below table. These results indicate the best performing models with the specified validation method on the overall dataset. Stratified K-fold splits followed by repeated cross validation methods yields the best performance along with training the model exhaustively without any bias.

Model	Validation Method	Accuracy(percentage)	F1 Score (percentage)
Long Short Term Memory(LSTM)	Stratified K fold splits, Cross Validation	85.35	85.01
Convolutional Neural Network (CNN)	Stratified K fold splits, Cross Validation	95.31	94.92
K Nearest Neighbor (KNN)	Repeated Stratified K fold splits, Cross Validation	69.42	67.46
Support Vector Machines (SVM)	Repeated Stratified K fold splits, Cross Validation	92.08	92.09

Table : Performance analytics of different models with the given dataset.

The confusion matrices of the above models are given below.



(a) CNN

[[84 2 0 2 0 0]
[7 32 7 0 0 0]
[5 0 57 5 2 5]
[5 0 0 51 3 2]
[2 4 0 0 46 12]
[0 0 4 0 1 70]]

(b) LSTM

Figure 4: Confusion Matrices

Confusion matrix - [[25 0 14 0 5 0]
[4 17 22 0 16 13]
[1 0 102 0 11 1]
[0 0 7 41 11 7]
[0 0 3 0 56 0]
[0 0 4 0 10 87]]

Confusion matrix - [[41 1 1 0 1 0]
[1 70 1 0 0 0]
[3 4 97 2 6 3]
[0 4 0 62 0 0]
[0 0 1 0 58 0]
[0 0 2 2 0 97]]

Figure 5: Confusion Matrices: KNN and SVM

5 Summary

The human activity recognition using different machine learning and deep learning approaches have been presented in this paper. The models performed significantly well on the given dataset.

The results indicate that the both classical machine learning and more recent deep learning models perform well for the human activity recognition. As the performance of different approaches can be improved with more data, SVM and CNN architectures perform well on time series data. LSTM comes a close to the performance but the architecture requires high computing resources compared to the above two in this specific scenario. The computing power is reduced especially in the case of classical machine learning approaches.

However, this needs to be thoroughly tested using a larger data set involving multiple participants and further research is required in this direction for a concrete model.

References

[1] Chen, Yuwen & Zhong, Kunhua & Zhang, Ju & Sun, Qilong & Zhao, Xueliang. (2016). *LSTM Networks for Mobile Human Activity Recognition. 10.2991/icaita-16.2016.13.*

[2] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. 2021. *Deep Learning for Sensorbased Human Activity Recognition: Overview, Challenges, and Opportunities. ACM Comput. Surv. 54, 4, Article 77 (May 2021), 40 pages.*

[3] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). *Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.*

[4] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. *A tutorial on human activity recognition using bodyworn inertial sensors. ACM Comput. Surv. 46, 3, Article 33 (January 2014), 33 pages. DOI: http://dx.doi.org/10.1145/2499621.*

[5] Yun Liu,Wendong Xiao, Han-Chieh Chao and Pony Chu. *Wearable Technologies, Sensor Technology Research Centre, University of Sussex, Brighton BN1 9RH, UK;*

[6] https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

A Appendix

For extra information or to get detailed understanding of the project, Please visit our Github Repository:

`https://github.com/kaushik4444/BasketballGame-ActivitiesClassifier`