Body-Rocking Behaviour Recognition

Akanksha Bhattacharyya
CSC Department
North Carolina State University
Raleigh, US
anbhatta@ncsu.edu

Deepayan Bardhan
ECE Department
North Carolina State University
Raleigh, US
dbardha@ncsu.edu

Sameer Watve
ECE Department
North Carolina State University
Raleigh, US
sswatve@ncsu.edu

I. METHODOLOGY

Sliding window method is used in the time series data for feature extraction. Values at multiple time steps can be grouped to form an input vector while the output corresponds to a specific label given to this vector. The features that were extracted were based on the paper by **Prof. Lobaton Edgar(2017)[1] Activity-Aware Physiological Response Prediction Using Wearable Sensors**. Some of the features include mean, covariance, skew, kurtosis and other frequency based features. The toolkits that were used were Scipy, Sklearn, Numpy and Pandas. The window size was kept to 3 secs which included 150 samples. The overlap of windows was taken to be 60 percent at the time of training. The results (cross validation set accuracy) were not affected much by changing window size to 5 seconds and/or changing overlap to 50 to 70 percent range. During testing also the values are grouped together. Size of each window is again 3 seconds (150 samples). But here sliding window =is 1 unit, that means 99 percent overlap between the windows as we need to predict output labels corresponding to each input.

It is clear from the graph that model 1 learned more properly, and thus we decided to go with model 2 which is CNN-LSTM architecture.

II. ARCHITECTURE

We have used two architectures for the time series classification. Following are those:

Architecture 1: CNN-LSTM
1D Convolution layer with kernel size = 2 and 32 filters and ReLU activation
MaxPooling layer with pool size = 2 LSTM layer with 500 neurons
LSTM layer with 200 cells Fully connected layer with 500 neurons.
Fully connected layer with 100 neurons.

Architecture 2: LSTM
LSTM layer with 500 neurons.
LSTM layer with 200 neurons.
Fully connected layer with 100 neurons.
Fully connected layer with 200 neurons.
The first model 1 has 2,033,597 trainable parameters while model 2 has 1,146,401 trainable parameters.

Model 1 took longer time to train, and the training vs validation accuracy and loss plots for 10 epochs is shown in Fig1 and Fig2 respectively.

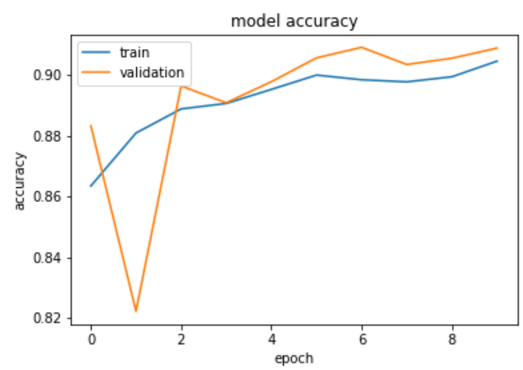

Fig. 1. Train and Validation accuracy for model 1: batch size = 128

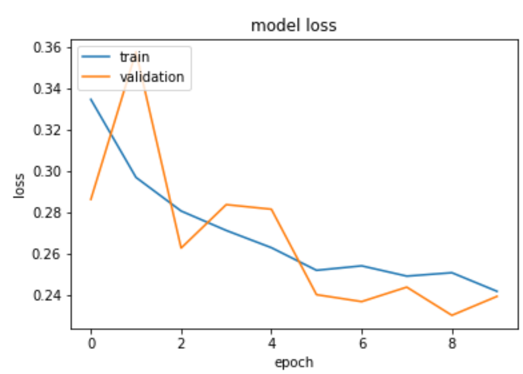

Fig. 2. Train and Validation loss for model 1: batch size = 128

On the other hand model 2 training vs validation accuracy and loss plots for 10 epochs is shown in Fig2 and Fig4 respectively.

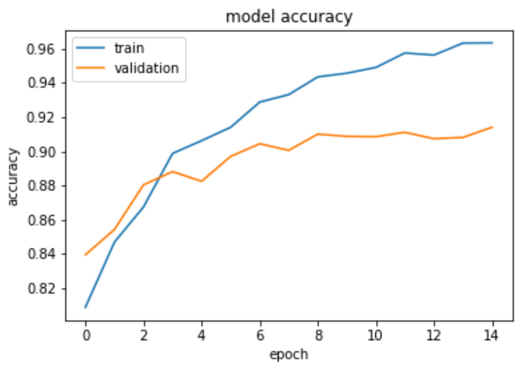

Fig. 3. Train and Validation accuracy for model 2: batch size = 128

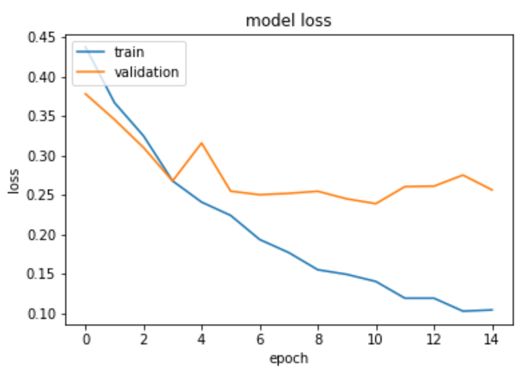

Fig. 4. Train and Validation loss for model 2: batch size = 128

III. MODEL TRAINING AND HYPER-PARAMETER SELECTION

The training datatsets consisted of 9 sessions out of which we used 7 for training and 1 for cross validation, 1 for testing. The following hyper-parameters were chosen to be optimized:

- batch size : [128, 256]
- learning rate : [1e-2,1e-3, 1e-4]

Number of epochs set to 10 as cross validation accuracy was observed stable beyond that point. It was observed from the grid search that the model with the following hyper-parameters produced the best accuracy and performance:

- batch size : 256
- learning rate : 1e-3

IV. EVALUATION

Test set accuracy is the error metric we considered while evaluating the model. The accuracy on the test set was 92.35 percent. The graph of accuracy and loss given by model is plotted as shown in figures 5 and 6. With the best model hyper-parameters, the model is trained on all 9 training sets and predictions were generated for the 4 test datasets provided. The predictions of the test data were saved in text format. b

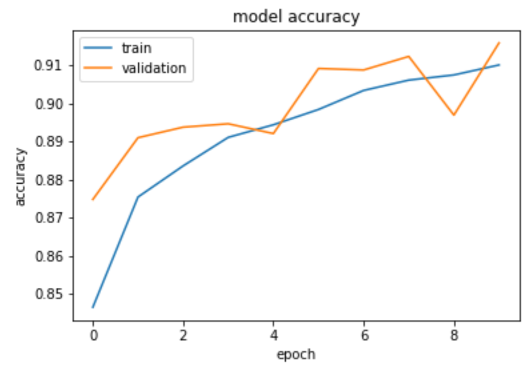

Fig. 5. batch size: 256 learning rate: 1e-3

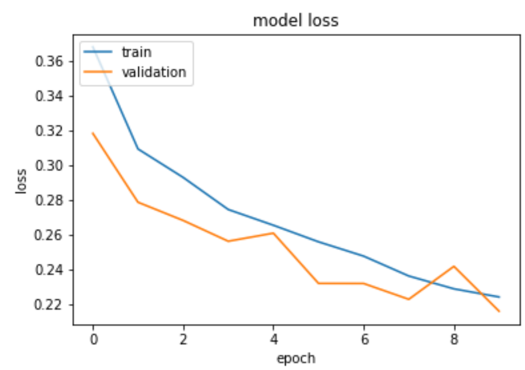

Fig. 6. batch size: 256 learning rate: 1e-3

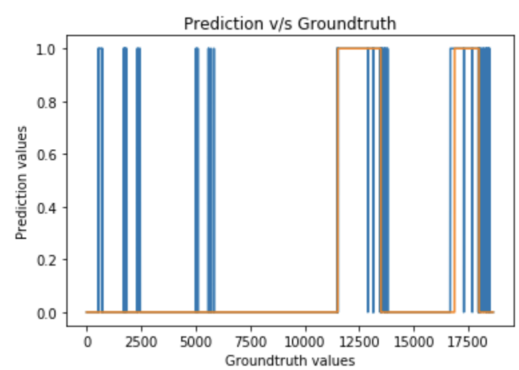

Fig. 7. Ground Truth vs Predicted result

V. CONCLUSION

Time series analysis using LSTM gives impressive results as compared to classical machine learning techniques like Random forest used in Project 3a. The hyperparamter dependence of model was studied.

REFERENCES

[1] Activity Aware Response Prediction using Wearbale Sensors, by Namita Lokare, Boxuan Zhong, Edgar Lobaton

[2] CNN Approaches for time series classification, by Lamyaa Sadauk

[3] Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep learning, MIT press, 2016.