Classifying User Activity with Machine Learning for Embedded Devices

An Huynh,[1] Arianna Burch,[1] Sofia Lee,[1] Mehmet Ergezer[1] and Marko Orescanin[2]
[1]Wentworth Institute of Technology
550 Huntington Ave, Boston, MA 02115
Tel: +1 (617) 989-4760
[2]Naval Postgraduate School
1 University Circle, Monterey, CA 93943
{huynha1, burcha1, lees13, ergezerm}@wit.edu, marko.orescanin@nps.edu

Abstract

The goal of this research was to develop a machine learning model that could classify user motion accurately and be deployed to a wearable device with limited storage and computational resources. The research team explored several machine learning and signal processing algorithms to classify common user activities using accelerometer and gyroscope signals and evaluated them based on their accuracy and efficiency. We achieved high accuracy (at least 90%) using support vector machines with the help of feature engineering and convolutional neural networks (CNN) using only raw data. We then optimized the CNN via pruning, stripping and quantization to reduce its size significantly (over 50x) and to enable for deployment.

Introduction

With the rapid increase in the popularity of wearable technology and smartphones, it is now possible to track a user's motion in real time. Many companies have leveraged this ability into their applications, some of which are targeted towards tracking a user's pace. However, a person's motion is not limited to their running speed as there is a wide range of activities a person can perform from walking upstairs to laying down. Many of the current user activity detection algorithms rely on video or image processing (Ye Liu 2015). However, live recordings of people are often not accessible due to computational or legality reasons. Given these circumstances, we proposed a solution to classify user activity via sensors present in many consumer electronics: gyroscopes and accelerometers.

This research was conducted by a team of three undergraduate students in Computer Science, and two Computer Science professors. We employed a publicly available time-series dataset to train end-to-end on convolutional neural networks (CNN)) as well as traditional supervised algorithms such as support vector machine (SVM) classifiers. We explored different signal processing algorithms to uncover activity patterns and features that can be leveraged towards prediction tasks. We then utilized the dataset and our generated features to train a SVM classifer. A one-dimensional CNN model was developed using the raw time-series data. The goal was to not only to determine which classification method produces the highest accuracy, but also one that we could deploy to a wearable device with limited storage and computational resources.

The paper is organized as follows: Section 2 introduces the dataset employed in the research. Section 3 compares the developed classification models and presents the generated features. Section 4 discusses the optimizations performed on the selected model to reduce its size. Finally, Section 5 states the conclusion.

Dataset

We employed a publicly available dataset where a group of thirty volunteers were recorded while performing six activities wearing a smart phone (Samsung Galaxy S II) on their waist (Anguita et al. 2013), (Dua and Graff 2017). The six labeled activities were: Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing and Laying.

Using the embedded accelerometer and gyroscope, the research team captured 3-axial (x, y, z) linear acceleration and 3-axial (x, y, z) angular velocity, at a constant rate of 50 Hz. The sensor acceleration signals, which has a gravitational and body motion components, were separated by body acceleration and gravity. In total, nearly 10,000 instances of activities were shared and split into 70% training and 30% testing sets.

Models and Feature Selection

Table 1 summarizes the overall performance of our models using SVM and CNN. Other machine learning algorithms were also tested; however, these two achieved the best results.

Model	Features	Feature Size	Accuracy (%)
SVM	Raw time and frequency domain & generated features & their polynomial combinations	2352	90.8
CNN	Raw time domain	1152	92.4

Table 1: Table comparing the type and amount of features used by the models for training. Average test accuracy is also listed.

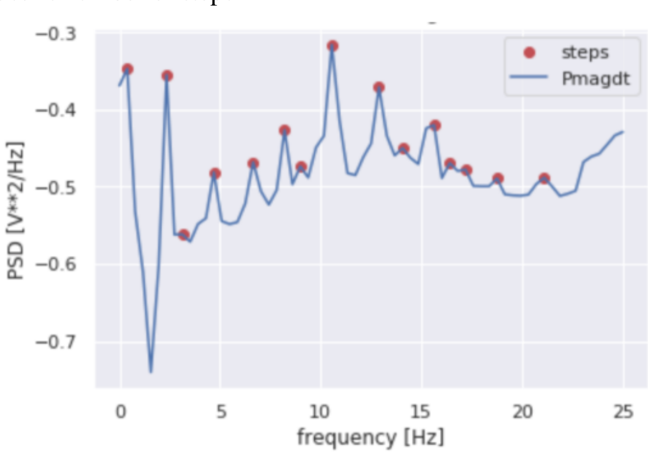
CNN architecture was implemented in Keras using 1D convolutions with filter sizes ranging from 64 to 128 and kernel size of 3 (Brownlee 2018). ReLU activation and batch normalization with 50% dropout and max pooling was employed between the layers. The model classified the raw time-series signals with three fully connected layers and softmax activation at top layer. CNN was selected for deployment over SVM to a wearable device as it produced higher accuracy of 92.4% using only the raw time-series data.

SVM required longer training time (75s) and was sensitive to tuning parameters than other traditional machine learning algorithms and CNN. Despite its longer training time, SVM correctly classified 75.4% using the raw time-series data. We were able to enhance its performance to 90.8% using feature engineering. Though the accuracy from SVM did not surpass CNN, we were able to demonstrate that the features we generated were able to improve the accuracy of our models by a substantial amount and that SVM's accuracy is on par with that of a neural network. For our SVM, we used the radial basis function kernel with a gamma of 0.001 and a C of 100.0.

Feature Selection and Generation

In order to increase the accuracy of our SVM model, we generated features from the raw time-series data (Ben D. Fulcher 2014). The data was first converted to frequency domain by calculating its power spectral density (PSD). To minimize the number of features, we added the peaks of the frequency data instead of the raw data. Figure 1 visualizes how we tracked the user's steps based on the peaks of the PSD. Furthermore, we calculated the mean, standard deviation, skew and kurtosis in both time and spectral domains. The last preprocessing step for the SVM was to calculate polynomial combinations of the these features to the degree of two.

Figure 1: PSD of the accelerometer signals in frequency domain (Hz). A peak finding algorithm was used to track the user's number of steps



CNN Model Quantization and Pruning

The designed CNN model's size is compressed by weight quantization and by removing extraneous parameters (pruning), and setting small weight values set to zero (stripping). This is implemented using TensorFlow Lites post-training quantization tools with polynomial decay starting at 50% and ending at 90%. Table 2 summarizes the results of CNN quantization in terms of model size and accuracy.

	Original	Pruned	Stripped
Model Size (MB)	1.6	0.51	0.03
Training Time (sec)	68	150	150
Test Accuracy	0.92	0.90	0.90
AUC	0.95	0.93	0.93

Table 2: Model size, training time and accuracy results through the compression stages of the CNN model.

Conclusions

In this research, we explored various machine learning models to classify human activities using accelerometer and gyroscope signals both accurately and efficiently. We wanted to not only determine the classification method with the highest accuracy, but also one that we could deploy to a small wearable device with limited storage and computational resources. Our models were trained on a time-series dataset and tested on prediction tasks (six common activities). In order to increase the accuracy of our SVM model, we generated various features from the training set and found that it increased the accuracy by around 20%. After comparing the performance of the CNN and SVM models, we selected the CNN model for deployment optimization to a wearable device as it provided the higher classification score, while requiring minimal expert knowledge or feature engineering. In addition, after performing pruning and quantization, we reduced the CNN model's size by over 50%, but as trade-off its accuracy reduced by 2%. The size-optimized CNN model provides a good comprimise between accuracy and computational cost and is ready for deployment on an embedded device. Future work includes testing the quantized CNN model in real-time in lieu of using a dataset.

References

Anguita, D.; Ghio, A.; Oneto, L.; Parra, X.; and Reyes-Ortiz, J. L. 2013. A public domain dataset for human activity recognition using smartphones. In *ESANN*.

Ben D. Fulcher, N. S. J. 2014. Highly comparative feature-based time-series classification.

Brownlee, J. 2018. How to develop 1D convolutional neural network models for human activity recognition. Accessed: 2019-07-30.

Dua, D., and Graff, C. 2017. UCI machine learning repository.

Ye Liu, Liqiang Nie, L. L. D. S. R. 2015. From action to activity: Sensor-based activity recognition.