Engineering

Taste of Research Summer Scholarship Program

Optimising Human Activity Classification using Deep Learning for Smart Devices

Author: Beatrix Leung

Supervisor: Salil Kanhere

Background and Motivation

- Rapid growth in the older population has resulted in the problem of supporting older adults in loss of cognitive autonomy, especially with those who wish to continue living independently.
- Smart environments have been developed to detect what occupants are doing, thereby detecting anomalies or deviations in an occupant's routine which could indicate a decline in abilities or even possible emergency situations.
- Smartphone applications with Human Activity Recognition (HAR) is an emerging alternative solution by integrating sensor networks with data mining and machine learning techniques to model a wide range of human activities.
- Deep learning networks have been chosen due to its effectiveness in discriminative feature learning. Specifically, both Long short-term memory (LSTM) and Convolutional Neural Networks (CNN) have been used, the latter of which exhibits greater accuracy in complex detection tasks.
- However, mobile phones cannot be overloaded with high computational requirements that undermine the performance of the phone and from a practical standpoint, memory available in mobile GPUs is insufficient to execute very deep CNNs, let alone in real time.
- Previous benchmarking studies have shown that Tensorflow has a better GPU memory management strategy on running convolutional neutral networks which has not yet been used to optimise the performance of human activity classifiers on smart devices.

Aims and Objectives

1. Evaluate the performance of different deep learning algorithms
2. Develop aa deep learning activity classifier using Tensorflow that can run on a smartphone
3. Optimise the classifiers and benchmark in terms of accuracy with small computational cost

Methodology

1. **DATA COLLECTION:** A public HAR dataset was used. Data was collected from a group of 30 volunteers with age bracket 19-48 years. Each performed six activities with a smartphone on the waist.
2. **FEATURE SELECTION:** Data was collected using an embedded accelerometer and gyroscope. Features chosen were:
   - Total Acceleration: Acceleration signal from the smartphone accelerometer (fig 1) in each of the axis in standard gravity units 'g'
   - Body Acceleration: The body accelerations signal obtained by subtracting the gravity from total accelerations
   - Body Orientation: The angular velocity vector measured by the gyroscope (fig 2) for each window sample.
3. **MACHINE LEARNING ALGORITHMS:** Information was extracted from the data collected by applying the machine learning algorithms including linear classifiers and deep learning (CNN and LSTMs)
4. **BENCHMARKING:** The different models were benchmarked according to accuracy and computational cost trade-off
5. **ANDROID APPLICATION BENCHMARKING:** The models were loaded onto an android application and its CPU performance monitored on a Google Nexus.

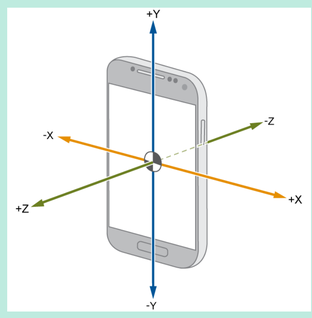


Fig 1. Android accelerometer

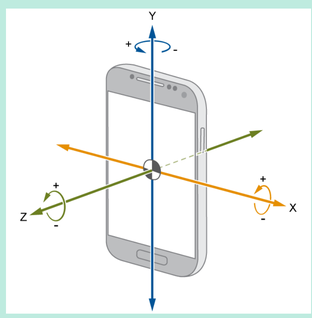


Fig 2. Android gyroscope

Results

1. LSTM vs CNN

After optimising both models, initial results showed CNN had much greater accuracy. Also highlighted the problem of differentiating sitting and standing
The table to the right shows the individual precision for each class for each model.
The confusion matrices below allows visualisation of performance for the LSTM (fig 3) and CNN (fig 4) models by showing percentage of total test data's predicted label as compared to its true label.

	LSTM	CNN
Walking	87.3%	97.2%
Walking Upstairs	93.4%	93.0%
Walking Downstairs	97.4%	97.4%
Sitting	81.7%	87.6%
Standing	84.6%	84.0%
Laying	95.0%	97.0%
TOTAL ACCURACY	**89.7%**	**92.6%**

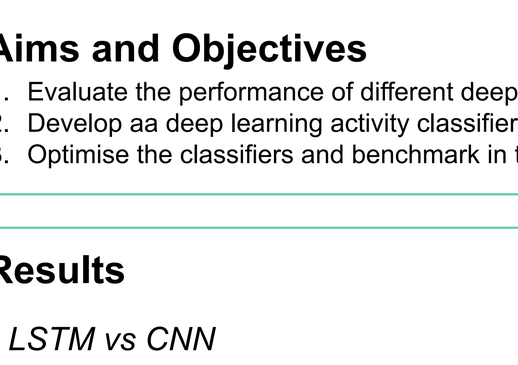


Fig 3. LSTM confusion matrix

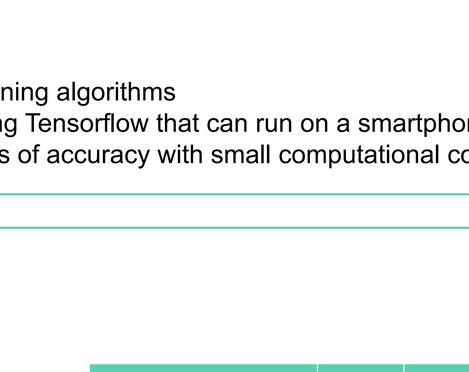


Fig 4. CNN confusion matrix

The sitting and standing data was separated and same process repeated to optimise classification of the two classes alone. Using the same CNN network with tuned regularisation parameter and learning rate, greatest accuracy of 86% was achieved. However, training all 6 classes data on the same model yielded only 92.1% accuracy.

2. Benchmarking

2.1 CPU usage on laptop

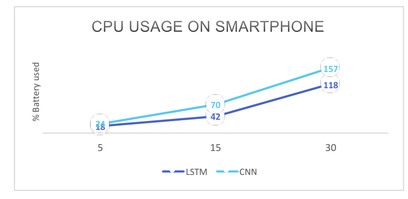
The trained models were benchmarked across 30 random test cases and CPU time averaged.

LSTM: 1.26s CPU time	CNN: 0.12s CPU time

2.2 CPU usage on smartphone

Next, using the GSam Battery Monitor app, the performance of a simple android app that ran data retrieved from the phone's in-built accelerometer and gyroscope on the trained models and displayed the output on screen was monitored.
Average time was taken over 30 runs.

CPU usage after:	LSTM	CNN
5 minutes	18 secs	24 secs
15 minutes	42 secs	1 min 10 secs
30 minutes	1 min 58 secs	2 min 37 secs

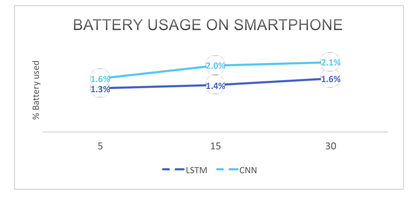


The computational load for both models appear to be of linear trajectory.

2.3 Battery usage on smartphone

The app ran in the background with no other apps running. Using the same GSam Battery Monitor app, the percentage of used battery (across all apps) used was monitored over 30 runs.

Battery usage after:	LSTM	CNN
5 minutes	1.3% battery	1.6% battery
15 minutes	1.4% battery	2% battery
30 minutes	1.6% battery	2.1% battery



2.4 Overall Performance of CNN

The CNN app of 92.6% accuracy was run overnight for 24 hours. In total it used 10 CPU minutes and consumed 2.1% of 27.4% total battery used by applications.

Conclusion

- Several methods of machine learning were used to develop models all of which were successfully operating on a smartphone. CNN was the most successful with 92.6% accuracy and consuming only 27.4% battery (including other basic apps) and using 10 CPU minutes.
- Using smartphones for HAR is a promising approach to assisting healthcare for elders and the mobility challenged. Fairly accurate classifiers can operate on smartphones without compromising it's main functionalities.
- However it also faces many challenges. The main ones being:
  - **Location Sensitivity** – due to the property of accelerometer's raw reading depending heavily on the sensor's orientation and position of the subject's body.
  - **Activity Complexity** – The transition period between activities and people performing multiple tasks at the same time are still problems. Hidden Markov Model could be a solution
  - **Energy & Resource Constraints** – Could become an issue as more activities are introduced. As it is, the CPU loading may be unsuitable for running in the background round-the-clock.