Online Human Activity Recognition using Low-Power Wearable Devices

G. Bhat*, R. Deb, V. Chaurasia
U. Y. Ogras

Holly Shill
Muhammad Ali Parkinson Center

ICCAD 2018
Outline

- Motivation
- Human Activity Recognition
  - Feature Set and Classifier Design
  - Online Reinforcement Learning with Policy Gradients
- Experimental Results
- Conclusions
- 15% of the world’s population lives with a disability
- 110-190 million people face significant difficulties in functioning
- *Intl. Parkinson and Movement Disorders Society Task Force on Technology:* 
  - Low-cost and small form-factor wearable devices offer great potential
  - Enabled by advances in low power sensors, processors, communications
Why Human Activity Recognition (HAR)?

- HAR identifies activities, such as walking, sitting, driving, jogging
- It is the first step to solutions for movement disorders

We have to know what the patient is doing to reach a conclusion

- HAR can provide valuable insight to health specialists
- Applications of HAR
  - Patient rehabilitation
  - Fall detection
  - Physical activity promotion
Why Online Learning on Wearable Devices

- Smartphones have been popular:
  - But, they are not appropriate
    - Some patients cannot even carry them
    - Large power consumption & charging requirements
    - Cannot provide real-time guarantees (e.g., sampling rate)
    - They are not designed for this purpose

Existing work on wearables and smartphones

<table>
<thead>
<tr>
<th></th>
<th>Offline</th>
<th>Online</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Collection</td>
<td>✔</td>
<td>❌ → ✔</td>
</tr>
<tr>
<td>Learning</td>
<td>✔</td>
<td>❌ → ✔</td>
</tr>
<tr>
<td>Inference</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

- Our solution
  - Tailored to the problem
  - Low power & Energy-harvesting
  - Adapt to new users and changing user conditions
Proposed Solution: Online Learning for HAR

- We use a wearable device to enable online learning for HAR
  - Uses a combination of motion and stretch sensors
  - First work to use stretch sensor for HAR

- **Our Novel Contributions**
  - Novel technique to segment the sensor data as a function of user motion
  - Online inference and training using reinforcement learning
  - Low power implementation on a wearable device
Outline

- Motivation
- Human Activity Recognition
  - Feature Set and Classifier Design
  - Online Reinforcement Learning with Policy Gradients
- Experimental Results
- Conclusions
Goals and Problem Statement

- Recognize *six common activities and transitions* between them
  - Achieve greater than 90% accuracy
  - Power budget in milliwatts

- These goals make HAR practical for daily use
  - Low power budget enables day-long operation using flexible batteries
Overview of Proposed HAR Framework

- **Segmentation**
  - Streaming stretch sensor data is processed to generate variable length segments

- **Feature Generation**
  - Accelerometer and stretch sensor data are processed to extract the features

- **Classifier Design**
  - Offline training of neural network using labeled segments

- **Reinforcement Learning**
  - Neural network weight updates using user feedback and policy gradient algorithm
Segmentation Algorithm

- **Need for variable length segmentation**
  - Fixed length segments may contain multiple activities
  - This makes it harder to label and classify

- **We use stretch sensor data to segment the activities**
  - Accelerometer is more noisy
  - In contrast, stretch sensor provides a clean data for segmentation

*Result: Non-uniform activity segments*
Detect local minimas in stretch sensor to define activity segments

Use five-point derivative to track trend

\[ s'(t) = \frac{s(t - 2) - 8s(t - 1) + 8s(t + 1) - s(t - 2)}{12} \]

Define the “Trend” as

- **Increasing** if \( s'(t) > 0 \)
- **Decreasing** if \( s'(t) < 0 \)
- **Flat** if \( s'(t) = 0 \)

Create a new segment when the trend changes from

- **Decreasing** to **Increasing**
- **Flat** to **Increasing**
Feature Generation

- Most of the prior work on HAR uses statistical features for activity classification
  - However, statistical features do not provide insight into the actual shape of data
- In contrast, we use DWT and FFT to get better insight
- Using this insight we generate the following features
  - Stretch Sensor: **16 FFT coefficients**, minimum and maximum values in the segment
  - Accelerometer: **32 DWT coefficients** for $a_x$, $a_z$ and body acceleration, mean of $a_y$
  - General features: Length of the segment and previous activity label
We use a parameterized neural network to classify activities

- Neural networks can be easily used for online learning

Neural network configuration

- One fully connected hidden layer
  - ReLU activation
- Fully connected output layer
  - Softmax activation

Probability of each activity is

\[
\pi(a_i|h, \theta) = \frac{e^{O_{a_i}(h, \theta)}}{\sum_{j=1}^{N_A} e^{O_{a_j}(h, \theta)}}
\]
Outline

- Motivation
- Human Activity Recognition
  - Feature Set and Classifier Design
  - Online Reinforcement Learning with Policy Gradients
- Experimental Results
- Conclusions
Classifiers shipped with the device can be trained using *known user data sets*

User patterns can change with
- Physical condition, age, gender, and, demographics

Even, condition of a given user may change over time

Classifiers learned offline must adapt to
- New users
- Varying conditions of its user

**Challenges**
- Online training can be computationally intensive
- Wearable devices do not have large storage area
Online Learning Preliminaries

- **State** ($\mathbf{X}$): Accelerometer and stretch sensor readings within a segment define the continuous state space.
- **Policy model** ($\pi$): The activity probabilities.
- **Action**: Activity performed in each segment.
- **Reward**: User provides the reward as a function of the classified action:
  - *If correct*: +1
  - *else*: -1

**Objective**: Maximize the total reward with respect to the classifier weights.
In general, all weights in the policy network are updated
- Useful when starting from an untrained network

We start with a trained policy network
- First few layers provide broadly applicable features
- Hence, we update only the output layer weights

Derived the policy gradient: $\nabla_\theta \pi(a_t|h, \theta_t)$

Found the weight update equation as

$$\theta_{t+1} \equiv \theta_t + \alpha r_t \frac{\nabla_\theta \pi(a_t|h, \theta_t)}{\pi(a_t|h, \theta_t)}$$

where $\alpha$: Learning rate, $r_t$: Reward
Motivation & Related Work

Human Activity Recognition
- Feature Set and Classifier Design
- Online Reinforcement Learning with Policy Gradients

Experimental Results

Conclusions
Experimental Setup

- **Wearable Device**
  - TI CC2650 MCU, InvenSense MPU
  - Stretchsense Stretch Sensor
  - MPU is sampled at 250 Hz
  - Stretch sensor at 100 Hz

- **Device Placement**
  - MPU is placed at the ankle
  - Stretch sensor is placed at the knee

- **User studies**
  - Total of 2614 segments from nine users
  - Five users used for offline training
  - Four users in online training

Our user data is available to public at OpenHealth page.
We use a neural network for supervised learning
- Needs to implemented on a device with 20kB of RAM

First, we fix the number of hidden layers to one
- Then, vary the number of neurons in the hidden layer

We swept the number of neurons
- Memory requirement increases linearly
- Accuracy saturates after four neurons

We choose four neurons in our NN
- Overall accuracy of 97.7 %
- Memory requirement of 2 kB
We analyze the confusion matrix for five users

All activities except Jump show an accuracy greater than 95%  
  – Jump shows higher variation among the users

Transitions have a lower accuracy  
  – This is acceptable as we can infer transitions from segments before and after

Confusion Matrix for All Activities

<table>
<thead>
<tr>
<th></th>
<th>Drive</th>
<th>Jump</th>
<th>Lie Down</th>
<th>Sit</th>
<th>Stand</th>
<th>Walk</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>D (155)</td>
<td>99.4%</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.6%</td>
</tr>
<tr>
<td>J (181)</td>
<td>0.00</td>
<td>93.4%</td>
<td>0.00</td>
<td>0.00</td>
<td>1.1%</td>
<td>3.9%</td>
<td>1.6%</td>
</tr>
<tr>
<td>L (204)</td>
<td>0.00</td>
<td>0.00</td>
<td>100%</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>S (394)</td>
<td>0.25%</td>
<td>0.25%</td>
<td>0.00</td>
<td>97.7%</td>
<td>0.76%</td>
<td>0.00</td>
<td>1.0%</td>
</tr>
<tr>
<td>Sd (350)</td>
<td>0.00</td>
<td>0.29%</td>
<td>0.00</td>
<td>0.00</td>
<td>98.6%</td>
<td>1.1%</td>
<td>0.00</td>
</tr>
<tr>
<td>W (806)</td>
<td>0.00</td>
<td>0.50%</td>
<td>0.00</td>
<td>0.00</td>
<td>0.62%</td>
<td>98.5%</td>
<td>0.37%</td>
</tr>
<tr>
<td>T (127)</td>
<td>0.00</td>
<td>3.1%</td>
<td>0.79%</td>
<td>2.4%</td>
<td>0.79%</td>
<td>2.4%</td>
<td>90.5%</td>
</tr>
</tbody>
</table>

Total number of windows with the corresponding activity
Reinforcement Learning for New Users

- Apply reinforcement learning for four new users
  - Never seen by the offline neural network

- Run the policy gradient update for a total of 100 epochs
  - Reward is given after every activity segment

- Accuracy improvement with online learning:
  - User 6: 74% → 91%
  - User 7: 89% → 94%
  - User 8: 86% → 96%
  - User 9: 60% → 91%

\[ \text{HAR algorithm adapts to new users} \]
Energy Consumption Analysis

- Prior studies do not report power & energy breakdown [1]
- Added test ports to our custom prototype
- Performed detailed power/performance/energy analysis

<table>
<thead>
<tr>
<th>Block</th>
<th>Exec. Time (ms)</th>
<th>Average Power (mW)</th>
<th>Energy (µJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sense Read / Segment</td>
<td>1,500.00</td>
<td>1.13</td>
<td>1,695.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compute DWT</td>
<td>7.90</td>
<td>9.50</td>
<td>75.05</td>
</tr>
<tr>
<td>FFT</td>
<td>17.20</td>
<td>11.80</td>
<td>202.96</td>
</tr>
<tr>
<td>NN</td>
<td>2.50</td>
<td>12.90</td>
<td>32.25</td>
</tr>
<tr>
<td>Overall</td>
<td>27.60</td>
<td>11.24</td>
<td>310.26</td>
</tr>
<tr>
<td>Communication BLE</td>
<td>8.60</td>
<td>5.00</td>
<td>43.00</td>
</tr>
</tbody>
</table>

Enables close to 60-hour operation with a 200 mAh battery

Wearable IoT devices offer great potential to enable interesting applications
- Health monitoring, activity tracking, gesture-based control

Presented a Human Activity Recognition framework
- Novel algorithm to segment data as a function of the activity
- Online inference and training using reinforcement learning
- Low power implementation on a wearable device

Data sets and source code will be made public
Comparison with Other Classifiers

- Comparison of our classifier to classifiers used by prior work

Our neural network classifier achieves compatible accuracy while enabling efficient online learning.