

# Smartwatch Gesture Recognition

Lucas Wojciechowski & Dane Carr & Ariel Weingarten  
Group 15

April 2, 2015

## 0.1 Abstract

## 0.2 Problem Statement and Solution Motivation

Wearable technology, such as smartwatches, are poised to become a ubiquitous consumer product. According to [businessinsider.com](http://www.businessinsider.com/smartwatch-sales-set-to-explode-2014-3), smartwatches sales are projected to reach 90 million units by 2018 (<http://www.businessinsider.com/smartwatch-sales-set-to-explode-2014-3>). Many user interface paradigms still need to be rethought for these devices to accommodate their small screens and limited input methods.

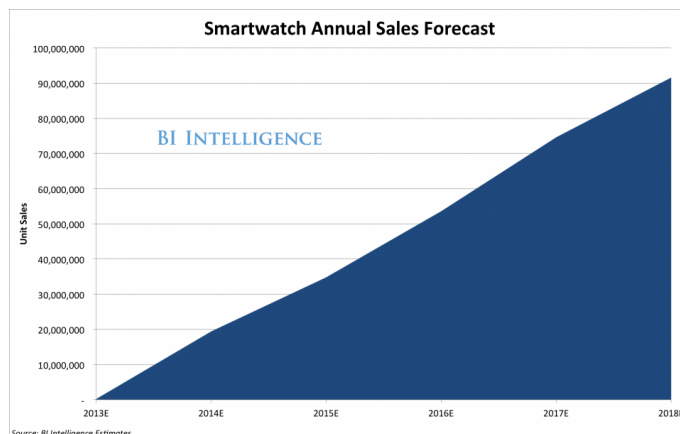


Figure 1: Smartwatch Annual Sales Projection

Currently, these devices rely mostly on touch input and voice control. Touch input is usually effective, but due to the small form factor, a significant portion of the screen is often obscured by the user's finger. Voice control can be used to complete more complex actions that might be tedious to complete using only touch input, but voice commands are not always practical. They may be less effective in noisy environments, or users might feel uncomfortable using them in public places due to issues with privacy and social acceptance.

Hand and arm gestures could be used to augment existing input paradigms. Though less expressive than voice commands, hand and arm movements are relatively private and they are a natural way to communicate with the environment. There are presently no publicly available gesture recognition solutions for off-the-shelf smartwatches, though theories and interface ideas have been prototyped with other sensors such as Microsoft Kinect or custom ultra-sound detection systems. The potential applications of this technology are far-reaching, from seamless control of Internet-of-things devices to accessibility for visually impaired users.

We have acquired a LG G Watch R and a Nexus 5 phone to record gesture data via a simple, custom-built Android application. Data collected from the watch's accelerometer will be used to train a neural network to classify a set of simple gestures. We will compare the accuracy and performance of our classifier against similar gesture recognition systems in the research community.

## 0.3 Theoretical Background Material

Our work is inspired by two works in particular. First, there is the famous "Learning Representations by Back-Propagating Errors" paper that first introduced the back propagation algorithm. Second, there is "Gestures without Libraries, Toolkits, or Training: A \$1 Recognizer for User Interface Prototypes". The former lays the groundwork for our solution and the latter provided motivation for the problem domain.

### 0.3.1 Backpropagation

Backpropagation has become a common method for training neural networks. Before its formulation, there was no way to provide a measure of accuracy for hidden layers in neural networks.

It is most often used in the supervised learning process as it requires the expected labeling for a set of inputs. Before the formulation of this algorithm there

### 0.3.2 \$1 Recognizer

s

## 0.4 Solution

We set out to build a system which would allow us experiment with smartwatch gesture recognition.

As a research platform, this system was designed to be extensible, simple, and easy to work with. As a potential consumer platform, this system was built to be portable and permanent enough to run natively on Andorid smartphones.

Due to the lack of prior art in this space, we had to build some fundamental infrastructure ourselves, such as a data collection system and data repository for the smartwatch gesture sensor data.

In the end, the system we built consists of the following components

1. a set of gestures that were easy to perform by users and easy to recognize by our machine
2. a data collection app collected smartwatch sensor data from users performing our gestures
3. a repository of training and testing data sets from the collection app
4. a custom neural network implementation designed for performance and portability on Android smartphones
5. a testing harness which replayed the data to train and test a neural network for gesture recognition
6. a neural network topology optimizer based on genetic algorithms

In the following subsections, we will discuss the requirements and implementation of each component in detail.

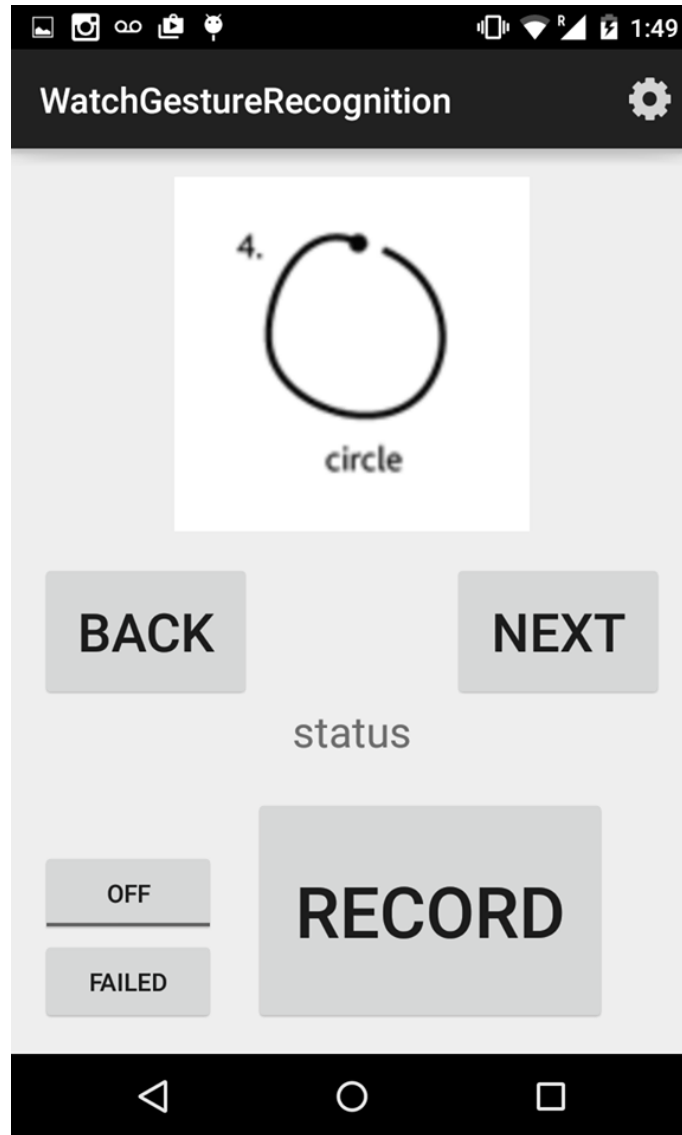


Figure 2: Screenshot of Android smartwatch sensor data collection app

#### 0.4.1 Gesture Set

#### 0.4.2 Data Collection App

#### 0.4.3 Repository of Testing and Training Data

#### 0.4.4 Neural Network Implementation

#### 0.4.5 Testing Harness

#### 0.4.6 Topology Optimizer

### 0.5 Results

#### 0.5.1 Codebase Metrics

#### 0.5.2 Training Performance

#### 0.5.3 Classification Performance

- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature*, 323, 533–536 [pdf]
- Wobbrock, J.O., Wilson, A.D. and Li, Y. (2007). Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '07)*. Newport, Rhode Island (October 7-10, 2007). New York: ACM Press, pp. 159-168.