What Am I Doing?

Robust Human Activity Detection with Smartphones

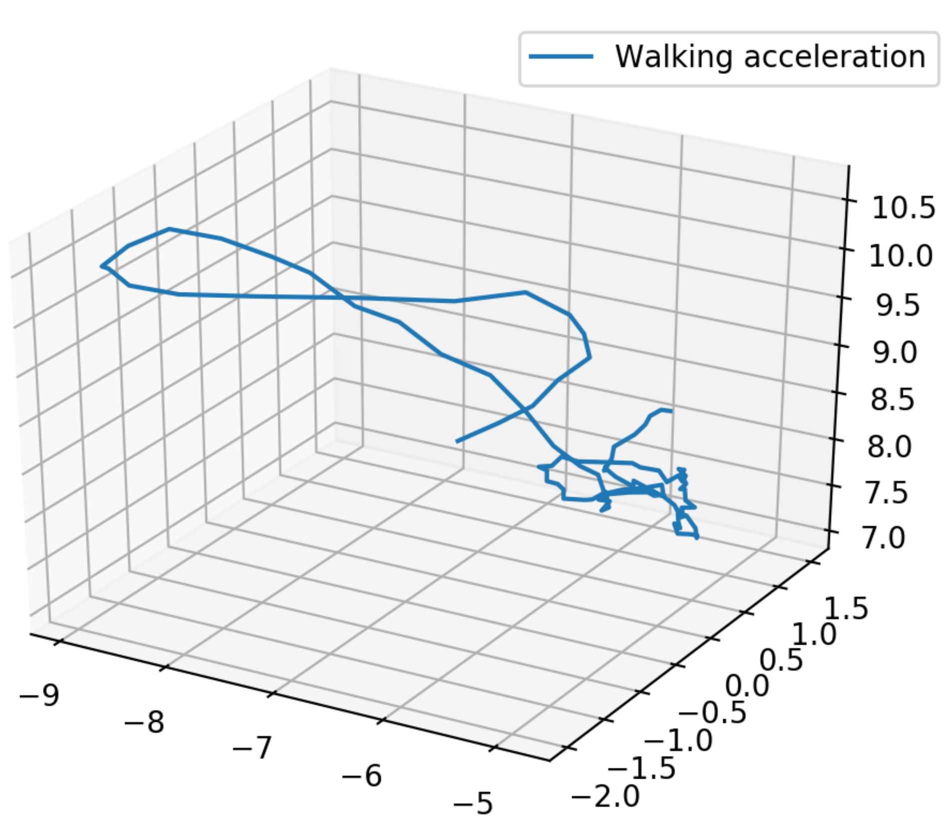
Eli Wu (eliwu) and Paa Adu (paa)
CS229 Fall 2017

Introduction

One of the most interesting areas of research with regards to HCI has been the problem of detecting what people are doing at a specific time - for example, detecting whether the person is biking, walking, sleeping, etc. This could potentially help with several tasks such as detecting exercising routines a la Google Fit, automatically bringing up a "cycling" UI displaying relevant information for cyclists, or turning on silent mode automatically when sleep is detected. In our project, we aim to ingest data from datasets such as the UCI Heterogeneity Activity Recognition Data Set and experiment on several different models to ultimately determine the best model.

Motivation

The wide-scale prevalence of smartphones and smartwatches has greatly improved the possibilities of human activity detection. Today, nearly everyone carries around a smart devices with sensors capable of measuring their movement (or lack thereof) and position, most notably accelerometers, gyroscopes, and magnetometers.

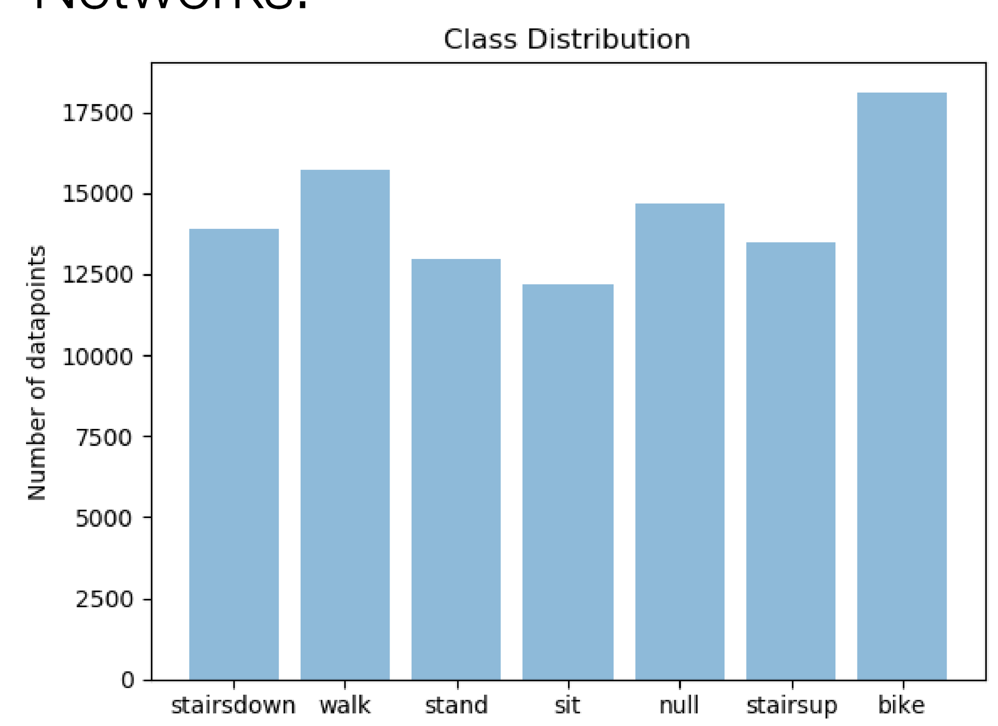


A plot of a sample of acceleration data gathered while walking.

This sensor data is already being utilized for applications such as step counting and localization. The application of sensor data towards activity detection has many interesting implications towards making our smart devices smarter.

Methodology

We evaluate four different models for classifying the sensor data. These are modified K-means prediction, Logistic Regression, Support Vector Machines, and Recurrent Neural Networks.



A plot of the distribution of the data classes.

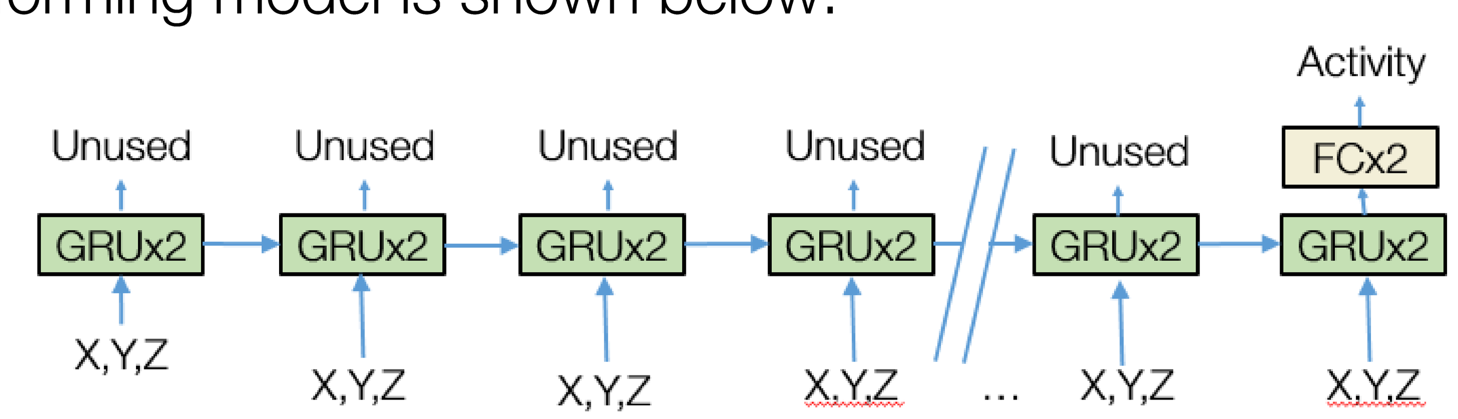
From the graph to the left, we can see that we can achieve above 26.9% accuracy by guessing bike; our models will need to achieve higher than this accuracy to prove that they work.

Modified K-means: Training data of the same class is clustered together. Then, one step of the k-means algorithm is run, creating new centroids for all the clusters. Once this is done, predictions are made on new data by clustering it to its closest centroid. Our performance metric for K-means is its ability to cluster new data to its corresponding cluster.

Logistic Regression: In order to make predictions over seven classes present in the dataset, we trained seven different logistic regression models, and picked the highest scoring model for prediction.

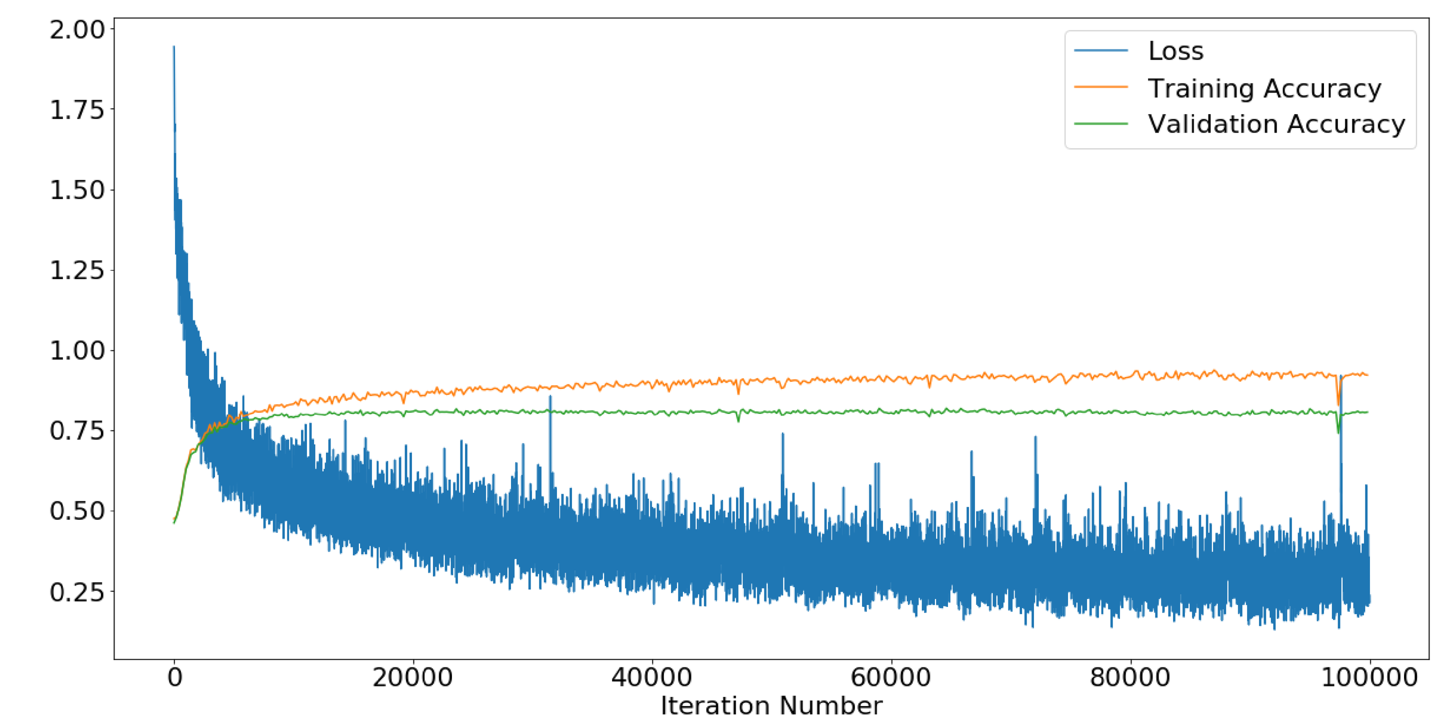
SVM: We utilized an SVM with a polynomial kernel. We trained seven different SVM models, and picked the highest scoring model for prediction.

Recurrent Neural Network: We tested several different recurrent architectures, using LSTM and GRU units; our best performing model is shown below.

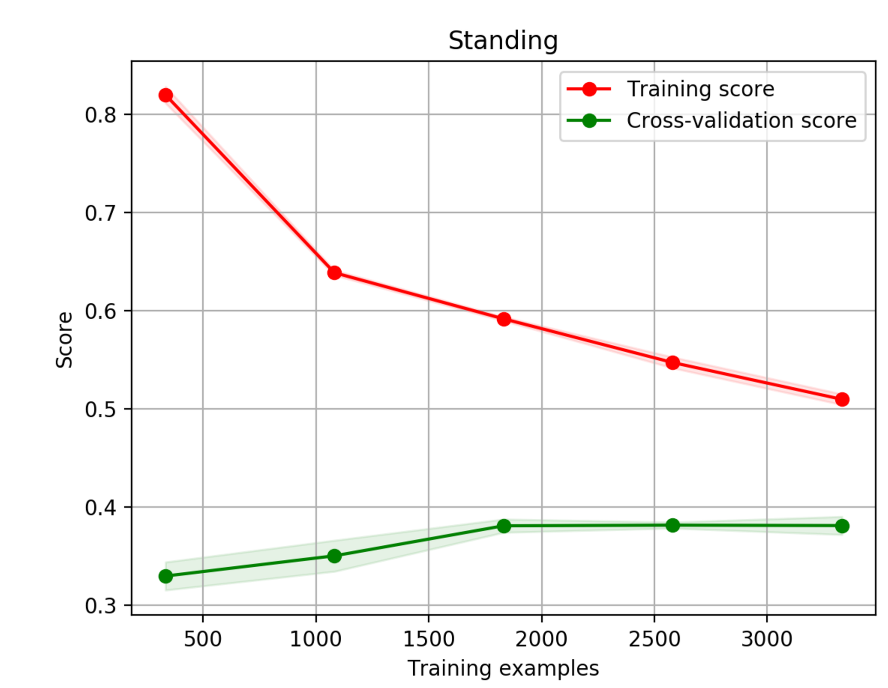


Results

Model	Train	Val	Test
Modified K-means	N/A	27.74%	27.22%
Logistic Regression	46.88%	34.42%	33.2%
SVM	65.23%	56.3%	55.6%
GRUx2, FCx2, Dropout	**90.86%**	**81.83%**	**80.79%**
LSTMx2, FCx2	72.14%	70.76%	69.83%



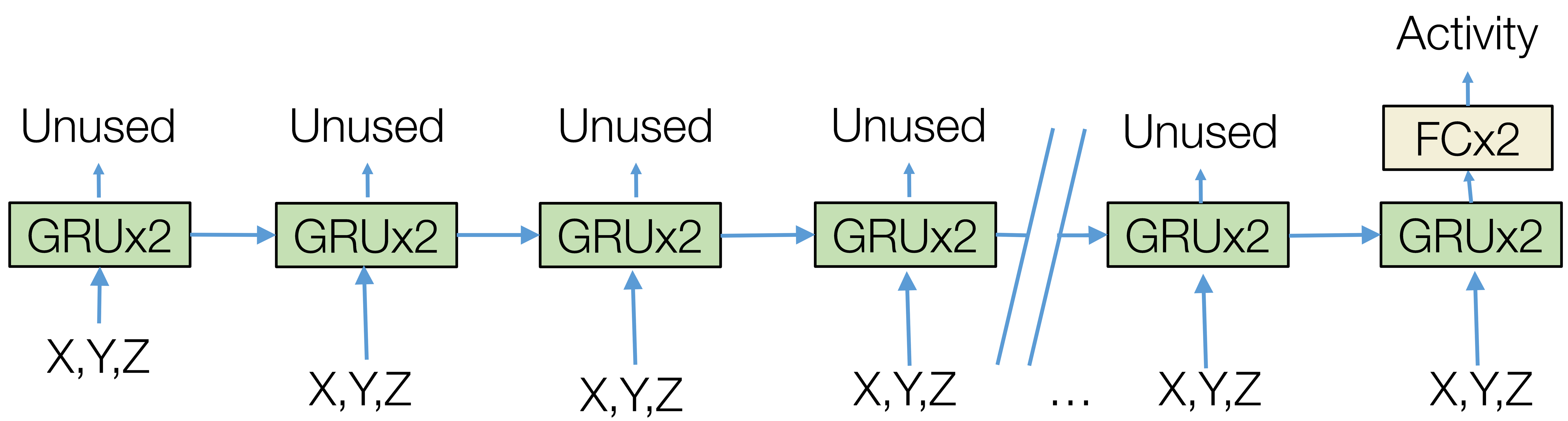
A plot of the training loss and accuracies for the RNN



A plot of the training loss and accuracies for Logistic Regression

Future Work

We intend to implement our GRU model on a simple Android app for real word testing. The app will ingest accelerometer and gyroscope data from a phone, and output the predicted activity. We hope this will be a tangible, real-world experiment to visualize our project.



Activity
FCx2
Unused
Unused
Unused
Unused
Unused
GRUx2
GRUx2
GRUx2
GRUx2
GRUx2
GRUx2
X,Y,Z
X,Y,Z
X,Y,Z
X,Y,Z
...
X,Y,Z
X,Y,Z