Datathon 5

Xiaoxuan Han, Hunter Ponzzebon, Gwen Eagle (Team 15)

Dalla Lana School of Public Health, University of Toronto

CHL 5230: Applied Machine Learning for Health Data

Dr. Zahra Shakeri

November 21, 2023

Introduction

Mobile health (mHealth) and the use of wearable technology that tracks physiological changes and physical activity have the potential to improve patient and healthcare system outcomes. Canada is a prime candidate for the adoption of mHealth into the toolkit of public health interventions, given the internet penetration of 93.8% and cellular connections accounting for 98.5% of the population. mHealth has the ability to increase healthcare accessibility in remote and underserved areas by reducing the reliance on in-person assessments. It also has the capacity to transform preventative care and increase patient engagement by encouraging patients to take a more active role in monitoring their health conditions and status through reminders, fitness trackers, and personalized health information. The plethora of long-term data collected by mHealth may help clinicians diagnose and better understand their patient's health trends, rather than relying on single point-in-time assessments. This may be especially helpful for patients who struggle to communicate their health concerns due to language barriers, lack of domain knowledge, anxiety, or disability. Overall, mHealth is a tool that can enable the delivery of personalized medicine to the Canadian population in an accessible and cost-effective manner.

Accurate tracking of physical activity levels is a feature of wearable technology that is important to both patients and clinicians for the delivery of personalized exercise and rehabilitation programs, remote patient monitoring, and early detection of health conditions. The objective of this paper is to build a machine learning model that can accurately ascertain activity type (multiclass outcome) and the 'presence of activity' (binary outcome) based on multidimensional time series data on linear and angular motion collected from wearable sensors placed on subject's left ankles and right lower arms. The secondary objective of this paper is to assess which model, recurrent neural network (RNN) or long short-term memory neural network (LSTMNN), has superior accuracy for predicting activity type and 'presence of activity'.

Data Engineering Process

To begin the data engineering process we first performed an exploratory data analysis. The dataset contained no missing values therefore no imputation was required. A binary variable was added to the dataset as an alternative target value called 'Activity_YN', where 0 represented no activity and 1 represented any activity (Activity 1 - 12).

In preparation for the neural network models, the data was processed by grouping by subject to ensure all motion measurements for the same individual remain within the same data set and subsequently splitting the data into 80% for training and 20% for test. The coordinates for the left ankle sensor and the right lower arm sensor were normalized and applied to the training and test datasets separately to prevent data leakage. Next, class imbalance in the training dataset was addressed for the data with Activity as the target and Activity_YN as the target. The data with Activity as the target was oversampled by entire patient sequences from the minority classes (activity 12 or 'Jump front and back x20') and undersampled from the majority classes (activity 0 or 'nothing'). The data with Activity_YN as the target was oversampled by entire patient sequences from the minority class (Activity_YN 1). A balanced data set was then created by combining minority class groups with the majority class group to create a balanced data set with a more balanced distribution of records from each class. Following this, sequence generation was performed to transform the input feature data, grouped by subject and identifying activity level as the target variable, into the same length of sequences for both training and test data and ensure recognition of time-series data.

Analysis

EDA: The dataset was first checked for any missingness as well as checking if missingness is above a specified threshold. Then, we printed the shape of the dataset, records of each patient, and counts of activity to better understand the data. The exploratory data analysis began by plotting a histogram for activity type to check for any skewness in the outcome, as well as histograms of the predictor motion variables by the activity class to understand their distribution. A histogram was plotted displaying the count of each activity type by subject. A correlation heat map was constructed to assess for multicollinearity between motion variables.

RNN: An RNN model was selected due to its unique ability to use internal state to process the sequential nature of the activity-based data. Different hyperparameters were explored and the final hyperparameters were defined as 256 hidden units in the RNN, with 13 classes, 100 epochs, a learning rate of 0.001, a batch size equal to 512, and a regularization term (lambda) equal to 0.0001 to prevent overfitting with Activity as the target. When Activity_YN was the target the hyperparameters were the same except for a learning rate of 0.00001 and a regularization term of 0.00001. Both datasets were prepared to be loaded in

batches for training and testing. The RNN architecture was defined with the following layers: an RNN layer with the input size classified as the train data with sequence generation and 256 hidden units, one dropout layer with probability of dropout equal to 0.5 to mitigate overfitting after the RNN layer, and two fully connected linear layers with an additional dropout layer in between (probability of dropout equal to 0.2). A Rectified Linear Unit activation function was applied to introduce nonlinearity into the network. The loss function was set to Cross-Entropy loss, which is appropriate for multiclass classification and the Adam optimizer was used to update the model parameters during training to minimize the loss.

The training loop was set to reduce the learning rate by 10% every 10 epochs, and for each epoch, the optimizer was created with the current learning rate. The model was set to training mode and predictions and losses were computed for a forward pass. Subsequently, backpropagation was performed to compute gradients and the weights and bias were updated based on the gradients computed. The model was switched between training and evaluation modes (without the computation of gradients or dropout behavior) to calculate training and validation accuracy for each epoch.

LSTMNN: The LSTMNN model was selected due to its capacity for enhanced memory. The process of building in the LSTMNN model differed from the RNN model only in the hyperparameters and the use of an LSTM layer as the first layer. The hyperparameters were defined as 256 hidden units in the LSTM model, 13 classes, 20 epochs, a learning rate of 0.001, a batch size equal to 512, and a regularization term equal to 0.0001 when Activity was the target. The hyperparameters were the same when Activity_YN was the target except the learning rate is 0.00001 and the regularization rate is 0.00001. Training and validation accuracy was similarly calculated and plotted for each epoch.

Findings

The longest length of observation by subject was 161,280 observations and the shortest length of observation was 17,726 observations, with zero missing observations across all 14 variables. The majority of motion variables had a mean of 0. All 9 subjects except for 1 performed all 13 activities, with the majority of observations corresponding to activity 0 (‘nothing’) and a minority of observations corresponding to activity 12 (‘Jump front and back x20’). The correlation matrix for the 12 motion variables revealed a positive correlation of 0.57 and 0.60 between gyro from the left ankle sensor and gyro from the right lower arm, and between acceleration from the right lower arm sensor and gyro from the right lower arm sensor, respectively.

Using “Activity” as the target, training accuracy for the RNN followed a trend of improvement, reaching approximately 85% by the 100th epoch, indicating the model learns to better fit the training data with each epoch. However, the validation accuracy remains low, consistently hovering around less than 20% across all epochs. The validation accuracy trend shows volatility with an overall plateau. The increasing training accuracy combined with the low and stagnant validation accuracy suggests that the model is performing significantly better on the training set than the validation set, providing evidence of overfitting. Using “Activity_YN” as the target, training accuracy for the RNN followed a similar trend of improvement, converging around 83% by the 100th epoch, however, the validation accuracy immediately began decreasing and then quickly hovering around 61-62% for the remainder of the epochs.

Using “Activity” as the target, the training accuracy for the LSTMNN model also continued to increase as the model learned from the data, reaching approximately 81% by the 20th epoch. However, similar to the RNN model, the validation accuracy remained very low, consistently hovering around 20% across each epoch. Thus, the LSTMNN model also demonstrates evidence of overfitting, given the plateaued validation accuracy despite the increasing training accuracy. Using “Activity_YN" as the target, the validation accuracy continued to trend upwards, reaching a near 83% accuracy by the 20th epoch, however, similar to the RNN model, the LSTMNN validation plateaued around 62% by the 20th epoch.

Conclusions

Overall, the RNN and LSTMNN models predict the specific and binary activity type with a higher degree of accuracy in the training dataset compared to the unseen validation dataset, as expected. For activity type, the RNN model demonstrated similar prediction accuracy, compared to LSTMNN (validation accuracy ~20%). The models for the ‘Activity_YN’ outcome outperformed those for activity type, with the RNN and LSTMNN models performing similarly (validation accuracy of ~62%). A high degree of overfitting occurred in all 4 models, potentially due to the models trying to predict a high number of classes based on a low number of subjects, as well as a high degree of class imbalance. Future directions include exploring different regularization techniques, hyperparameter tuning, and using larger sample sizes to improve the accuracy.

Author's Contribution: All group members contributed equally to the completion of this project.

Github Code Slideshow