# Tennis Shot Classifier

## Table of Contents



---

## Introduction

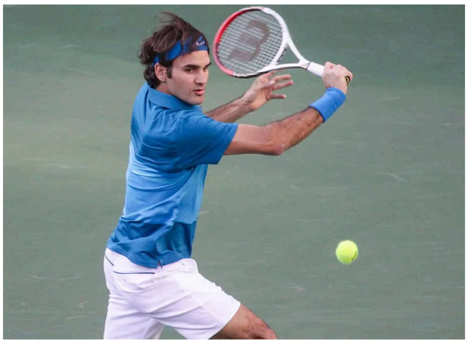
Our project involved the classification of different tennis shot types and running around the tennis court. For this purpose, our data was acquired from MATLAB's mobile application featuring sensors that collected acceleration, orientation, angular velocity and position data where our cellphones simulated the role of the tennis racket.

The shot types classified were the *forehand* and the *backhand*.

The **forehand** is a type of shot in tennis and other racket sports, made by swinging the racket across one's body with the hand moving palm-first. It is a groundstroke where the ball has bounced before it is struck. The inner part of the hand (palm) faces the ball during a forehand shot. In tennis, it is the opposite of a backhand shot (https://en.wikipedia.org/wiki/Forehand).

The **backhand** is a stroke used in most racket sports, such as tennis, table tennis and pickleball. It is a way of hitting the ball in which the back of the hand holding the racket is turned in the direction you want to hit the ball. The term refers to a groundstroke, except in the phrase backhand volley. The backhand is a shot made with the back of the hand turned in the direction of movement (https://en.wikipedia.org/wiki/Backhand).

Roger Federer's forehand



Roger Federer's backhand

# Selected Sensor Data

The motion sensors sample rate was 10Hz. The collected data for the training were about 30-40 seconds long. The initially recorded sensor data were:

1. Acceleration (10Hz samples)
2. Orientation (10Hz samples)
3. Angular Velocity (10Hz samples)
4. Position (defaults to 1Hz samples)
   a. Latitude
   b. Longitude
   c. Speed
   d. Course
   e. Hacc

We decided to use the data from the x, y, z axes for acceleration, orientation and angular velocity in order to classify the forehand and backhand as they are correlated to the task at hand. The differences in speed acquired from the position were deemed too small and inaccurate to use for classification in the case of the shots. The speed would be crucial in telling apart running from shots, but was not used due to the fact that the position data had a different length compared to our other data.

# Project Files

## Data

The data files listed below included the sensor data for each shot, the running or walking around the court and finally, the map of the court, where basically the recorded sensor data of the individual consisted of the person standing for approximately 5 seconds on the 4 edge of the court area and walking to the next corner, one by one in order to create a mapping of the space and where the player moves. The sensor data types in the files were the same for every case.

### Data files

forehand2.mat

backhand.mat

run_walk.mat

map.mat

## Program

### Main Program

TennisClassifier.m

### Functions

extract_features.m

### Unfinished Program (Court Mapping)

maptest.m

# Feature Extraction

We decided to split the data for every 20 samples and then calculate the mean value and standard deviation for either acceleration, orientation or angular velocity in the x, y, z axes. This way, we created a matrix with 18 columns (3 types of data x 3 axes x 2 statistical tests). Current code only uses the acceleration data. The sample window is overlapping, meaning that at each feature vector corresponds to the last 20 samples at a given moment. This could be extended to an 18 valued vector by using the angular velocity and orientation axis data.

# Normalization

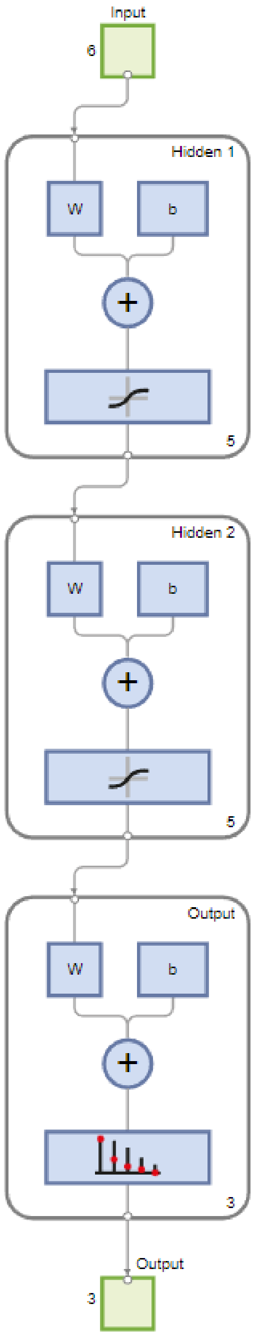
As sensor data vary in values they are normalized to have a zero mean and a standard deviation of one to ensure homogeneity and higher success rate, since features with very high values are not dominant over others.

# The Classifier Model

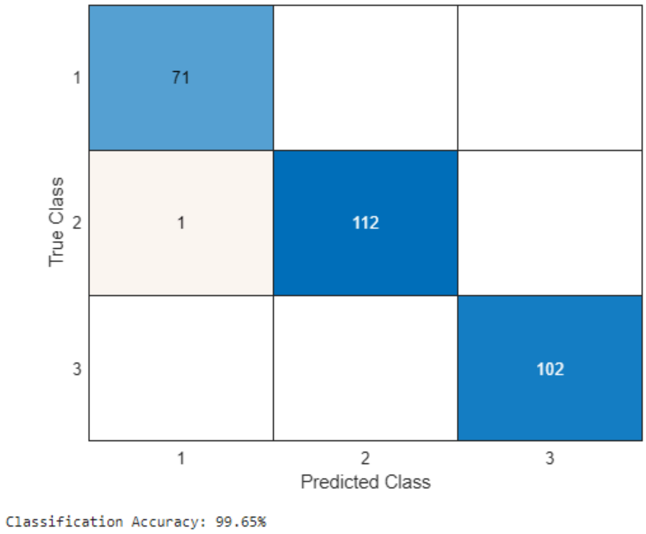
We used an MLP (Multi-Layer-Perceptron) neural network model for our of 6 inputs, 2 hidden layers of 5 neurons each and a three valued output layer. The amount of neurons on the hidden layers were picked randomly. Data are split into 70%, 15% and 15% for training, testing and validation of the MLP model. This is done for learning the model, the actual testing is done on data we didnt pass to the model training to use them as "new" inputs.

We trained our model using the default MLP gradient updates and passed the test data on our model for predictions. The model showed great accuracy but it should be tested on more than one people for generallity.

*Note: The testing data had to be split before normalization, as normalization is done only on learning data. We didn't have enough time to adjust our code.*




Model Architecture




Confusion Matrix and Accuracy

# Possible additions

## Court mapping

A possible addition to the project would be using the longitude and latitude in order to find the corners of the tennis court and thus create a mapping of the space were the player moves, creating a live history of the game's events. While, we attained the longitude and latitude of said positions, we were unable to create the simulation on MATLAB. A preliminary example is found on maptest.m which we didn't have time to build.