Smartphone User Classification

-ML Project-

Cutic Eugen

# Description

The task at hand is to classify smartphone users based on a data set containing accelerometer recordings in the 3D space (x, y, z), using an ML algorithm.

# The chosen model

After trying an SVM, a neural network with 2 hidden layers, and a CNN, the one which performed best was the CNN. Its arhitecture is detailed below.

## Arhitecture

Here we have a keras generated plot of the model. Input goes through convolution(100 filters, kernel size 10), then max pooling with window size 3, then convolution again, followed by global average pooling and in the end there are 2 fully connected layers (with 128 and 64 nodes respectively) leading to the output layer.

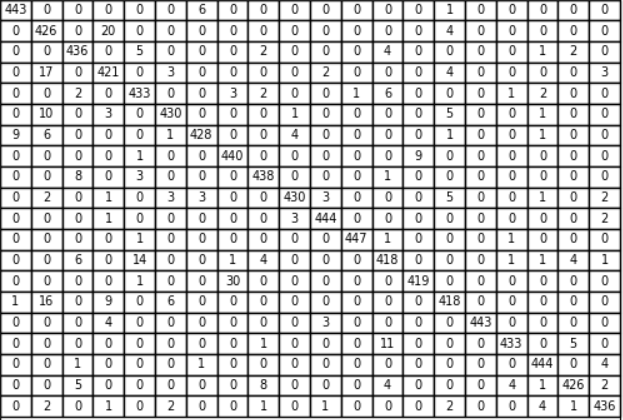
Before the dense layers there is a droput layer for regularization. Also, the 2 dense layers use l1 kernel regularizer and l2 activity regularizer.

The convolutional layers, as well as the hidden dense layers use the relu activation function, while the output layer uses softmax. Also, the model uses adam optimization with 0.001 learning rate.

## Evaluation

The model has been evaluated on a 0.3 data validation split, considering validation loss and validation accuracy over 100 epochs and this has indicated that it would be optimal to train for only 25 epochs, to avoid over-fitting.

Also, the mean accuracy for a 3-fold cross validation was 0.9603333473205566 and the confusion matrix for the predicted values on the validation set can be found below.



# Other models

Other models considered and their test accuracies: SVM(0.7 C value, rbf kernel) – 0.775; SVM(10 C value, rbf kernel) – 0.795; a neural network similiar to the chosen model, but without convolution, having only the 2 hidden layers and the output layer – 0.829.

# Bibliography

Tensorflow official documentation

Scikit-learn official documentation

To construct the convolutional part of the model, a tutorial was followed which explained the concept on similar data (accelerometer recordings) and which can be found here: <https://blog.goodaudience.com/introduction-to-1d-convolutional-neural-networks-in-keras-for-time-sequences-3a7ff801a2cf>