Toe-Walking Detection

Faaris Khilji

This document goes over the methods used to detect toe walking versus normal walking

Sensor placement:

Initially I recorded the raw data and trained the model using a basic ANN. I recorded 238 timesteps worth of data for each sample of walking or toe-walking. Each timestep contains six features being acceleration x,y,z and gyro x,y,z.

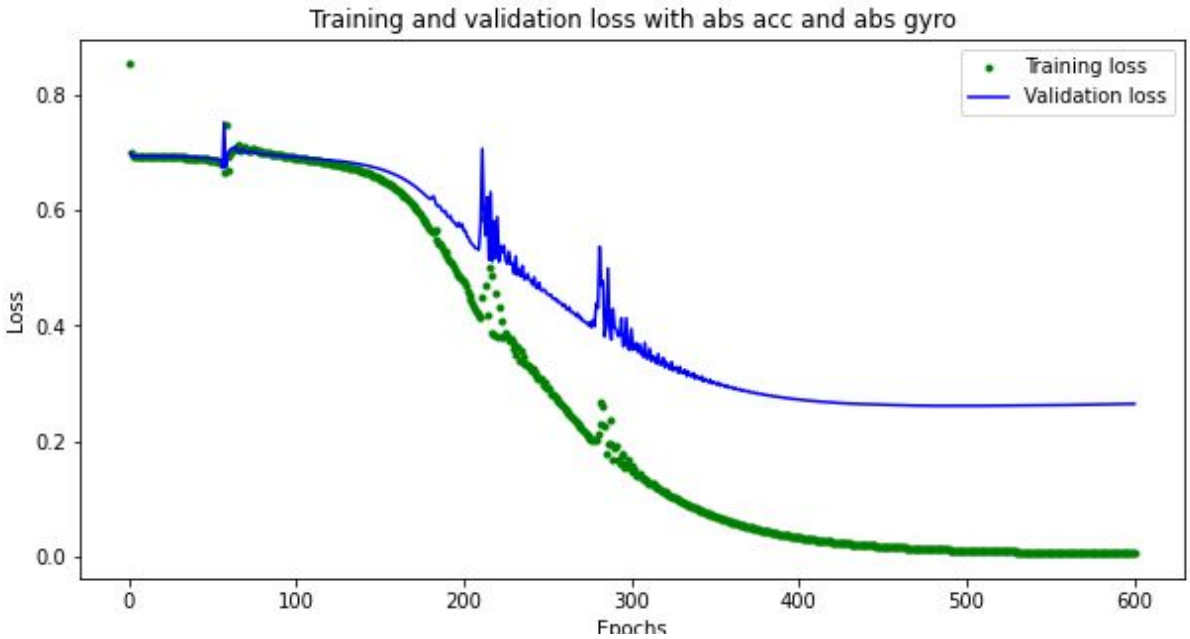
Below the code for the model is pictured:

```
model = tf.keras.Sequential()
model.add(tf.keras.layers.Dense(100, activation='relu'))
model.add(tf.keras.layers.Dense(50, activation='relu'))
model.add(tf.keras.layers.Dense(15,activation='relu'))
model.add(tf.keras.layers.Dense(NUM_GESTURES, activation='softmax'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Utilizing this model I trained the neural network and achieved an accuracy of 82% when testing it out live.

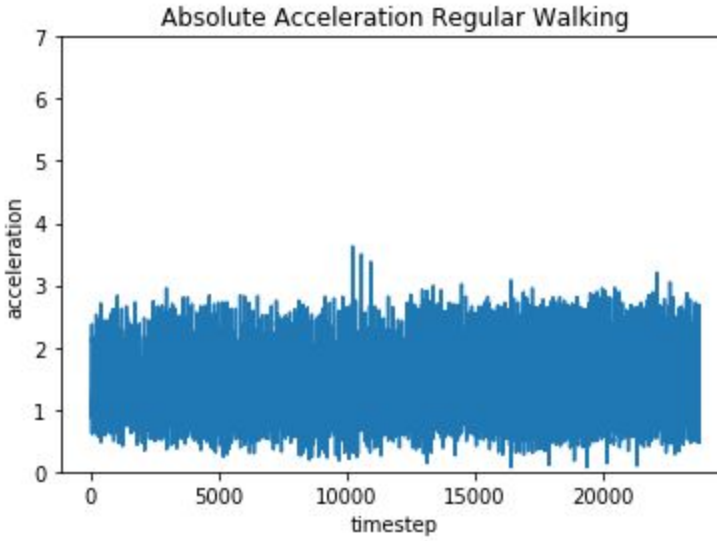
```
[[20  0]
 [ 3 17]]
              precision    recall  f1-score   support

           0       0.87      1.00      0.93        20
           1       1.00      0.85      0.92        20

    accuracy                           0.93        40
   macro avg       0.93      0.93      0.92        40
weighted avg       0.93      0.93      0.92        40
```

Confusion matrix and classification report for first model

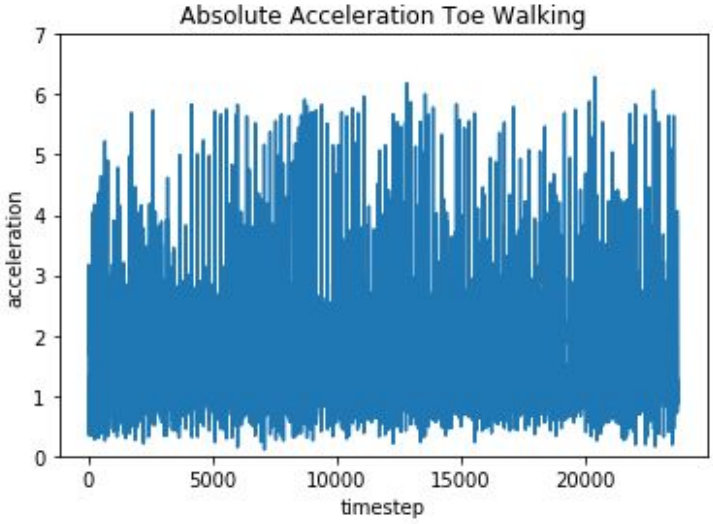


Improving the Model

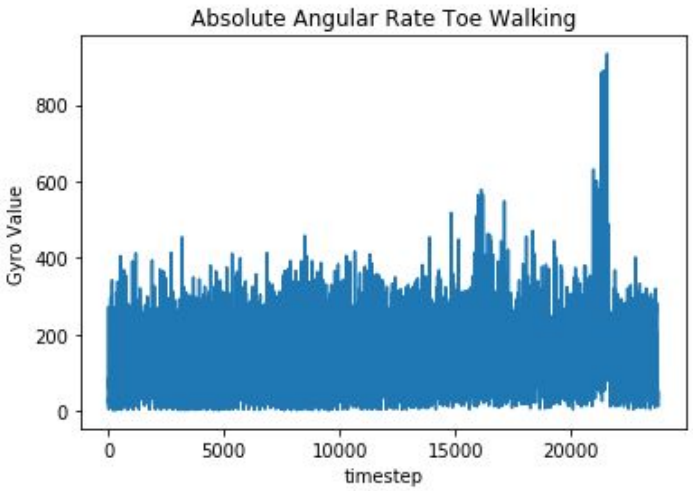
To improve the model I wanted to extract more features from the raw data.

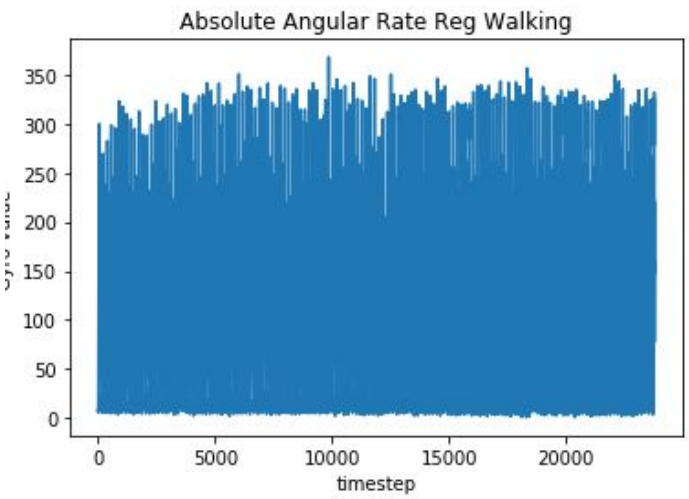



```
Max is: 3.6277687357382638
Min is: 0.06721606950722424
```




```
Max is: 6.292196119003285
Min is: 0.10246950765959599
```

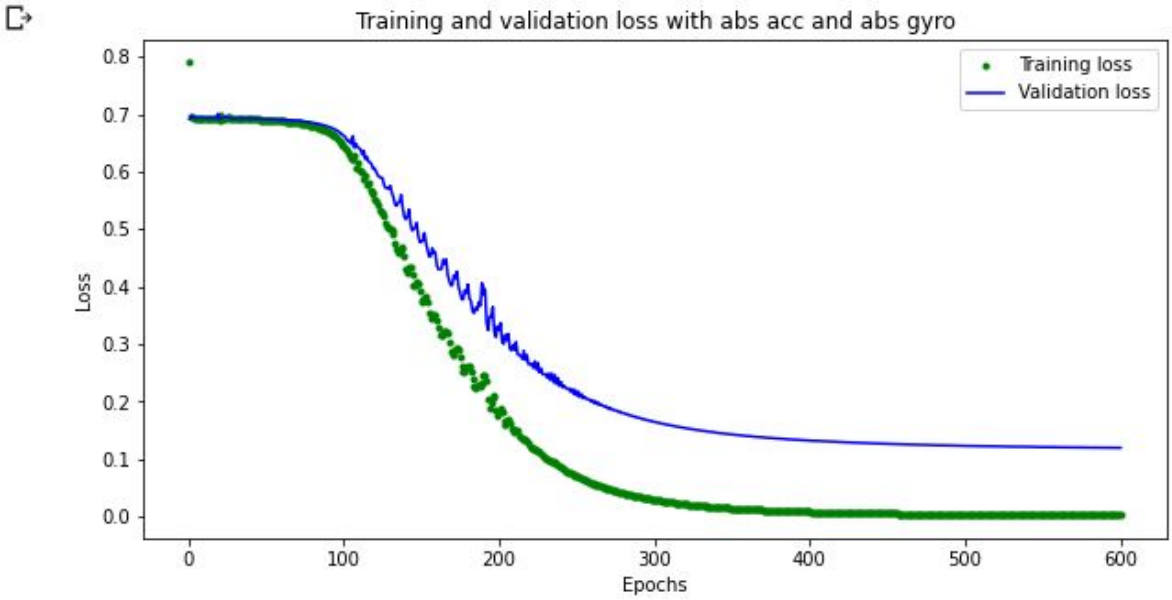





```
Max is: 933.6217185011284
Min is: 1.1013836752013353
```

```
ax is: 367.7676605494289
in is: 0.7569900924054422
```

I added these two features to the end of each line of data resulting in 8 features total to be passed into the model.




```
[10.0, 5.0]
```

```
[[20  0]
 [ 0 20]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        20
           1       1.00      1.00      1.00        20

    accuracy                           1.00        40
   macro avg       1.00      1.00      1.00        40
weighted avg       1.00      1.00      1.00        40
```

When testing this model out I did not encounter a time where it predicted incorrectly.

How Data was Split:

Type	# Samples
Training	160
Validation	20
Test	20

For each sample there are 238 timesteps and 6 or 8 features. In total this means there are 1428 or 1904 data points passed into model.