PROJECT REPORT

HUMAN ACTIVITY RECOGNITION USING SMARTPHONE DATASET

Adarsh Baghel 180001001
Jeevan Reddy 180001039
Ubaid Shaikh 180001050

Under the Guidance of

Dr Aruna Tiwari

Computer Science and Engineering

May 16, 2021

Introduction:

Smartphones are the most useful tools of our daily life and with the advancing technology, they get more capable day by day to meet customer needs and expectations. To make these gadgets more functional and powerful, designers add new modules and devices to the hardware. Sensors have a big role in making smartphones more functional and aware of the environment thus most smartphones come with different embedded sensors and this makes it possible to collect vast amounts of information about the user's daily life and activities.

Accelerometer and gyroscope sensors are amongst these devices too. The accelerometer has been standard hardware for almost all smartphone manufacturers. As its name suggests, an accelerometer measures the change in speed; not the speed itself. Data retrieved from the accelerometer may be processed in order to detect sudden changes in movement. Another sensor that has been standard hardware for smartphones is the gyroscope which measures orientation by using gravity. Signals retrieved by gyroscope can be processed to detect the position and alignment of the device. Since there is a meaningful difference of characteristics between data retrieved from these sensors, many features could be generated from these sensor data to determine the activity of the person that is carrying the device. Classification of smartphone user activities has been focused on in different studies.
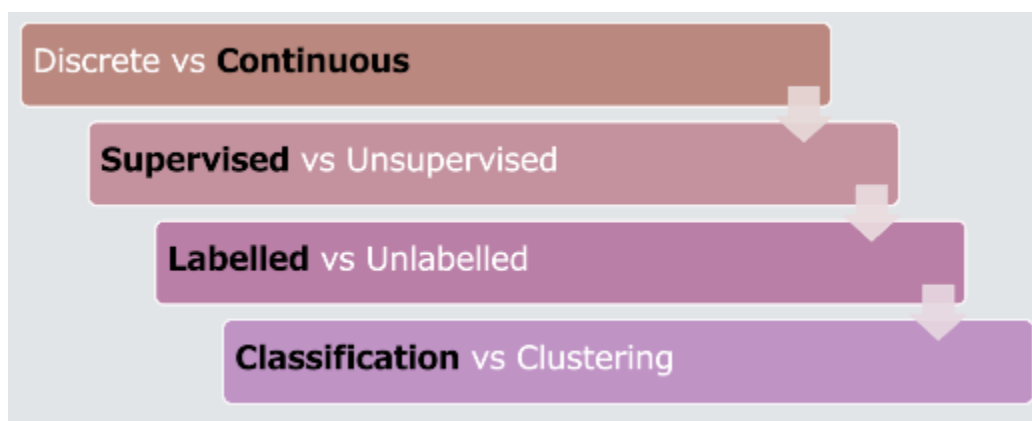
Description of Dataset:

The Human Activity Recognition database was built from the recordings of 30 study participants performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. The objective is to classify activities into one of the six activities performed.

The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING UPSTAIRS, WALKING DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers were selected for generating the training data and 30% for the test data.

The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low-frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain.

Dataset Source:

https://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions

Data Set Characteristics:	Multivariate, Time-Series	Number of Instances:	10929
Labeled:	Yes		

Feature description:

|	count
fBodyAcc	79
fBodyGyro	79
fBodyAccJerk	79
tGravityAcc	40
tBodyAcc	40
tBodyGyroJerk	40
tBodyGyro	40
tBodyAccJerk	40
tBodyAccMag	13
tGravityAccMag	13
tBodyAccJerkMag	13
tBodyGyroMag	13
tBodyGyroJerkMag	13
fBodyAccMag	13
fBodyBodyAccJerkMag	13
fBodyBodyGyroMag	13
fBodyBodyGyroJerkMag	13
angle	7
subject	1
Activity	1
Output	1

Train Dataset distribution
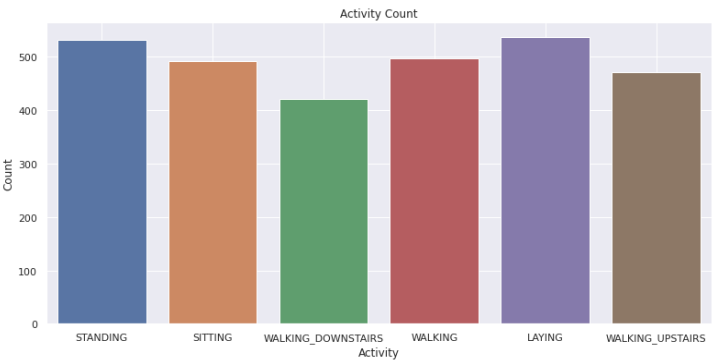
Test Dataset distribution.

Overview of Dataset:

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tBodyAcc-mad()-Y
0	0.288585	-0.020294	-0.132905	-0.995279	-0.983111	-0.913526	-0.995112	-0.983185
1	0.278419	-0.016411	-0.123520	-0.998245	-0.975300	-0.960322	-0.998807	-0.974914
2	0.279653	-0.019467	-0.113462	-0.995380	-0.967187	-0.978944	-0.996520	-0.963668
3	0.279174	-0.026201	-0.123283	-0.996091	-0.983403	-0.990675	-0.997099	-0.982750
4	0.276629	-0.016570	-0.115362	-0.998139	-0.980817	-0.990482	-0.998321	-0.979672

angle(X,gravityMean)	angle(Y,gravityMean)	angle(Z,gravityMean)	subject	Activity
-0.841247	0.179941	-0.058627	1	STANDING
-0.844788	0.180289	-0.054317	1	STANDING
-0.848933	0.180637	-0.049118	1	STANDING
-0.848649	0.181935	-0.047663	1	STANDING
-0.847865	0.185151	-0.043892	1	STANDING

TSNE: Activity Visualisation
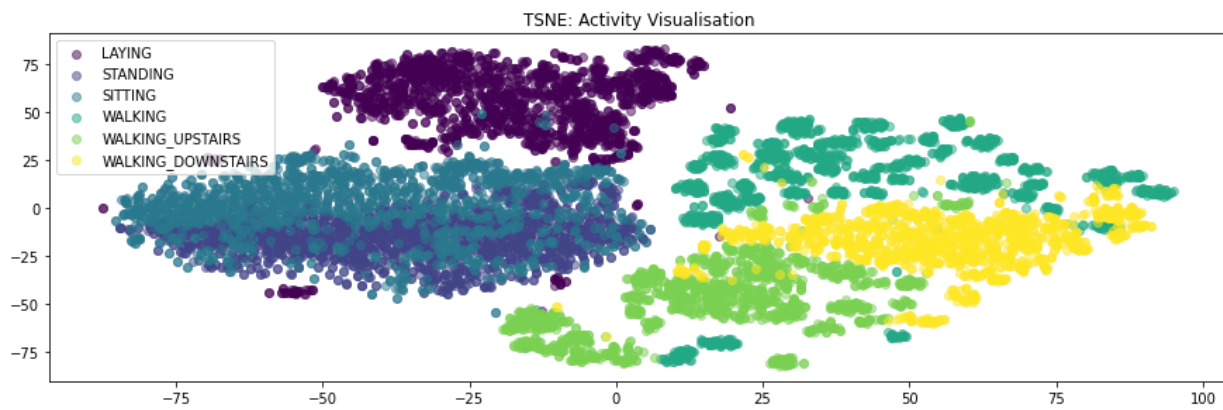
Learning Methods

Supervised machine learning is used to recognize activity from dataset records. Different supervised machine learning models designed using different classification approaches.

Designed models are first trained with training data that consists of %80 of the total dataset and then tested with the rest. Classification precision of models is tested and observed using 5-fold cross-validation.
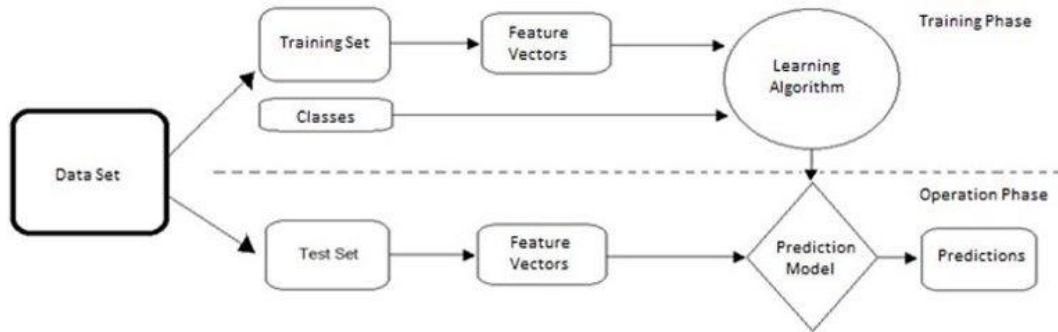
Artificial Neural Network

An artificial neural network (ANN) is a computational model consisting of interconnected artificial neurons (or nodes) that are inspired by biological neural networks. ANNs are able to model complex relationships between inputs and outputs or to find patterns in data.
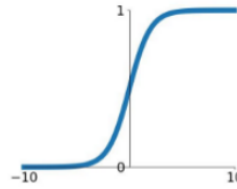
In this project, we use a class of ANN called multilayer perceptron (MLP) as a classifier as illustrated in the following figure. The backpropagation algorithm is used in the training process.
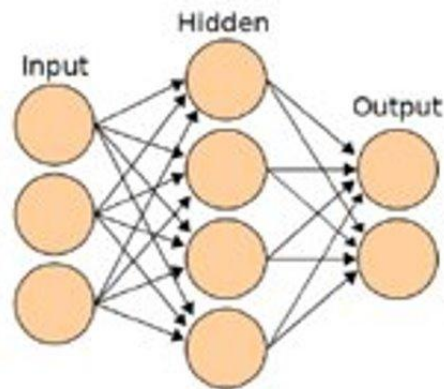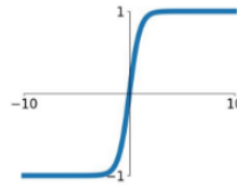
Activation Functions Used:
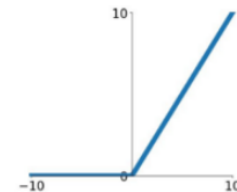
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$

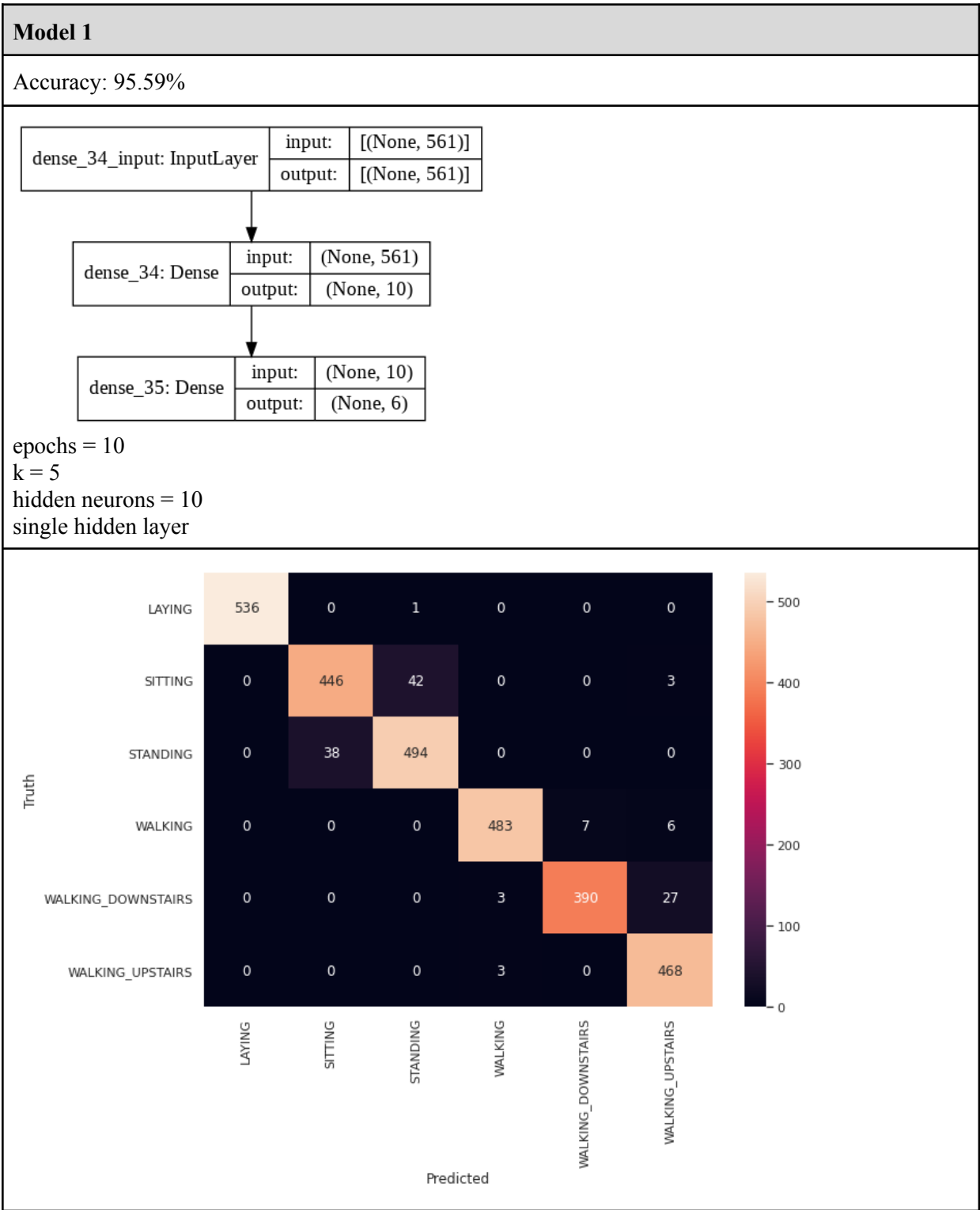**tanh**

$\tanh(x)$

ReLU

$\max(0, x)$

Sigmoid: The Sigmoid function is a smooth function and is continuously differentiable. The biggest advantage that it has over step and linear function is that it is non-linear. This is a very important feature of the sigmoid function. It essentially means that when we have multiple neurons having sigmoid function as their activation function – the output is non linear as well. The function ranges from 0-1 having an S shape. We are using the Sigmoid Activation function in the output layer of all our neural network models

Tanh: tanh is also like logistic sigmoid but better. The range of the tanh function is from (-1 to 1). tanh is also sigmoidal (s - shaped). The advantage is that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in the tanh graph. The function is differentiable. The function is monotonic while its derivative is not monotonic.

Relu: The ReLU function is the Rectified linear unit. It is the most widely used activation function. We are using this in the hidden layers of 1st five neural network models. The main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time. If the input is negative it will convert it to zero and the neuron does not get activated.
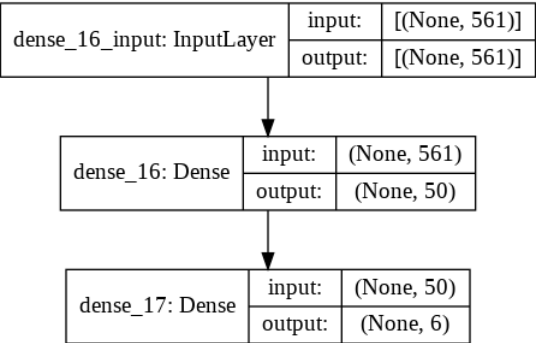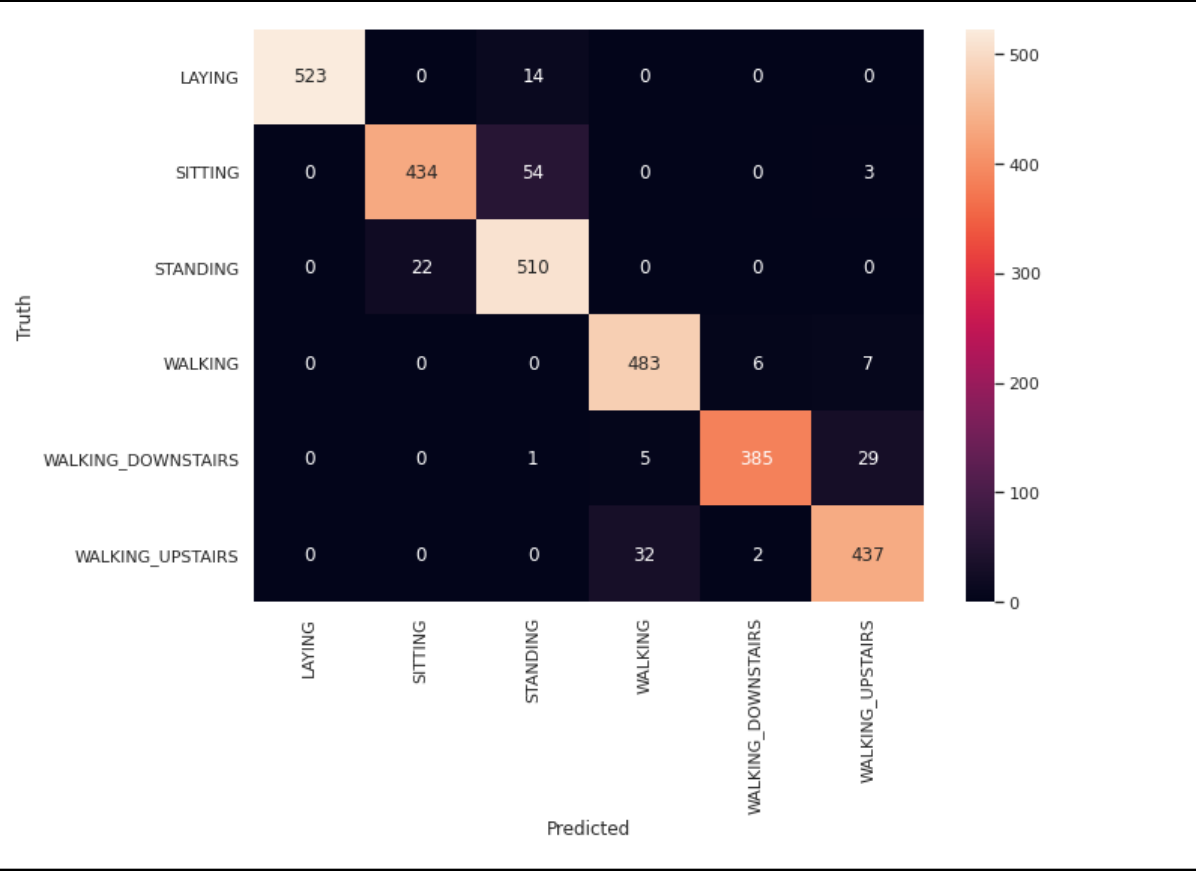
Results and Analysis:

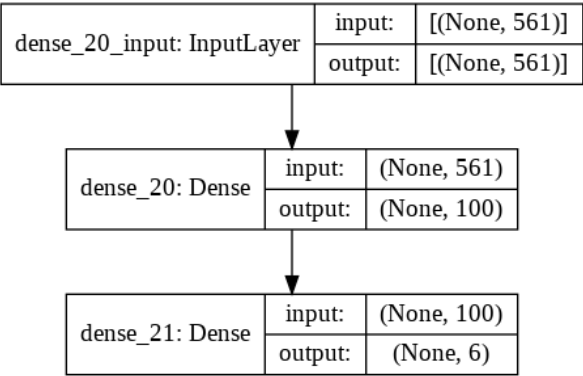| Model 1 |
|---|
| Accuracy: 95.59% |

| dense_34_input: InputLayer | input: | [(None, 561)] |
| | output: | [(None, 561)] |

| dense_34: Dense | input: | (None, 561) |
| | output: | (None, 10) |

| dense_35: Dense | input: | (None, 10) |
| | output: | (None, 6) |

epochs = 10
k = 5
hidden neurons = 10
single hidden layer

Model 2

Accuracy: 94.06%

| dense_16_input: InputLayer | input: | [(None, 561)] |
| | output: | [(None, 561)] |

| dense_16: Dense | input: | (None, 561) |
| | output: | (None, 50) |

| dense_17: Dense | input: | (None, 50) |
| | output: | (None, 6) |

epochs = 10
k = 5
hidden neurons = 50
single hidden layer

Model 3

Accuracy: 94.77%

| dense_20_input: InputLayer | input: | [(None, 561)] |
| | output: | [(None, 561)] |

| dense_20: Dense | input: | (None, 561) |
| | output: | (None, 100) |

| dense_21: Dense | input: | (None, 100) |
| | output: | (None, 6) |

epochs = 100
k = 5
hidden neurons = 100
single hidden layer
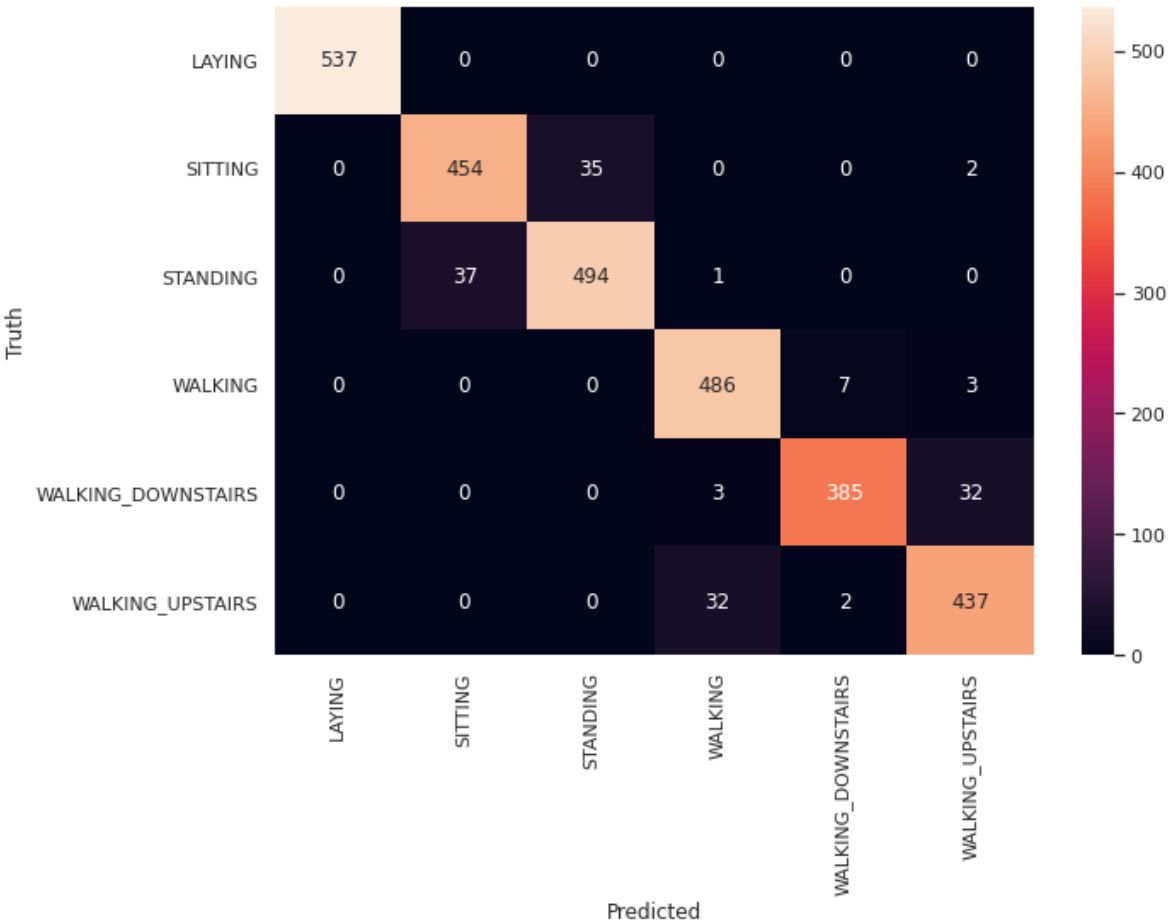
Model 4

Accuracy: 94.40%

| dense_24_input: InputLayer | input: | [(None, 561)] |
| | output: | [(None, 561)] |

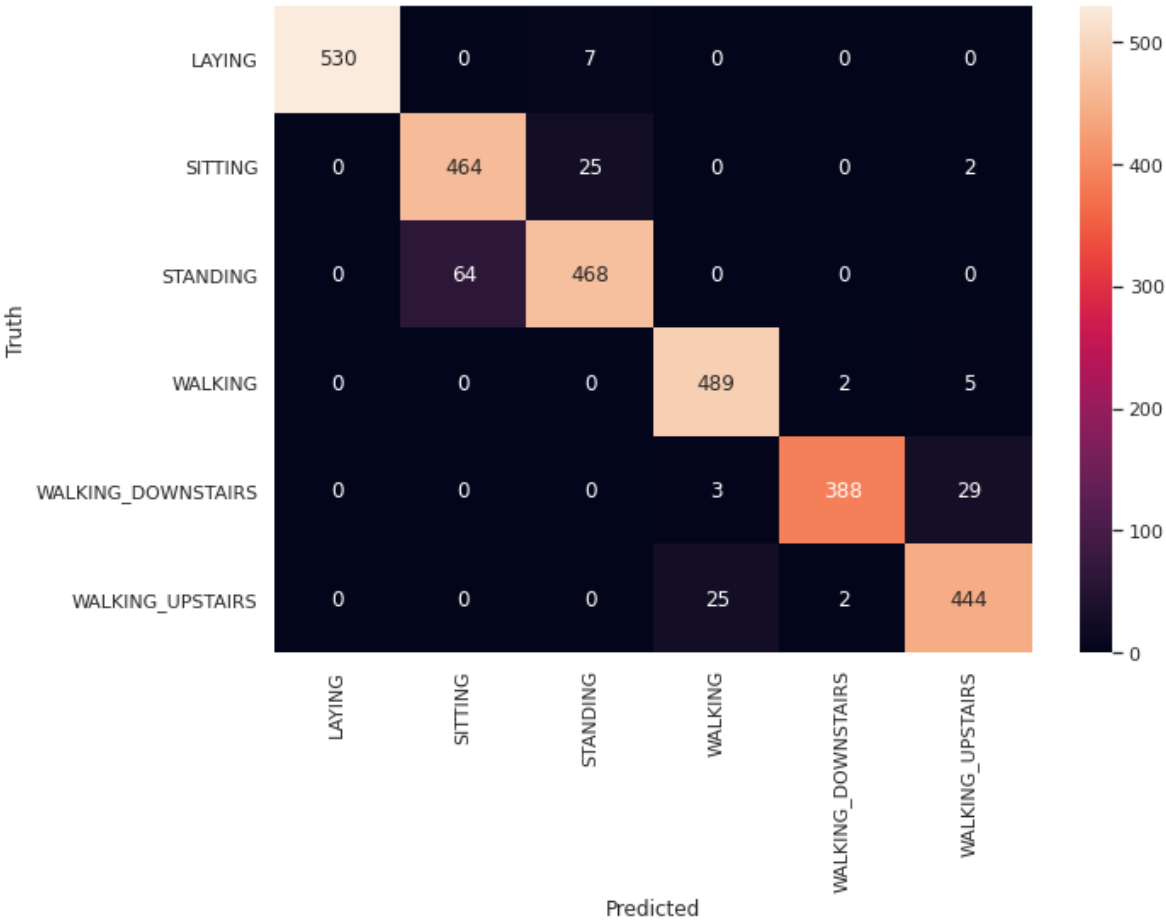| dense_24: Dense | input: | (None, 561) |
| | output: | (None, 100) |

| dense_25: Dense | input: | (None, 100) |
| | output: | (None, 6) |
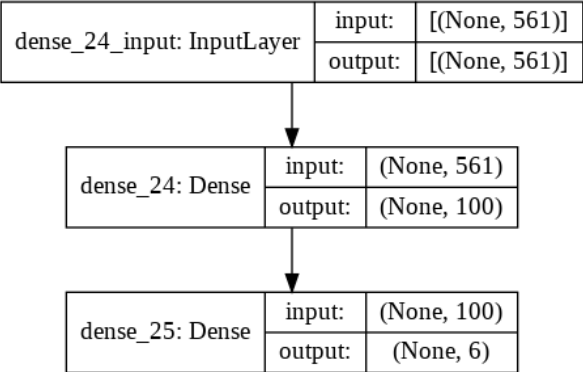
epochs = 100
No cross fold validation
hidden neurons = 100
single hidden layer

Model 5

Accuracy: 95.45%

dense_26_input: InputLayer	input:	[(None, 561)]
	output:	[(None, 561)]

dense_26: Dense	input:	(None, 561)
	output:	(None, 100)

dense_27: Dense	input:	(None, 100)
	output:	(None, 50)

dense_28: Dense	input:	(None, 50)
	output:	(None, 6)
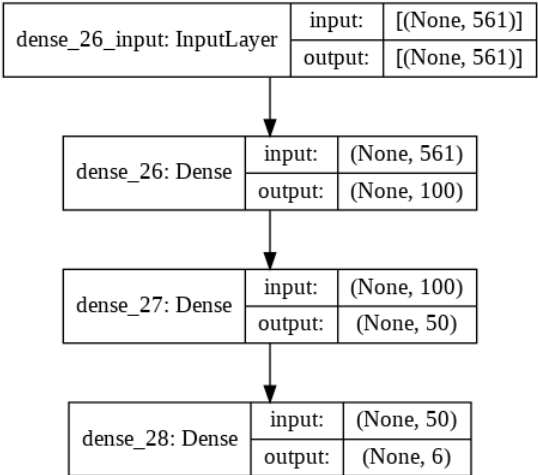
epochs = 100
No cross fold validation
hidden neurons = 100, 50
two hidden layer

Model 6

Accuracy: 95.08%

dense_32_input: InputLayer	input:	[(None, 561)]
	output:	[(None, 561)]

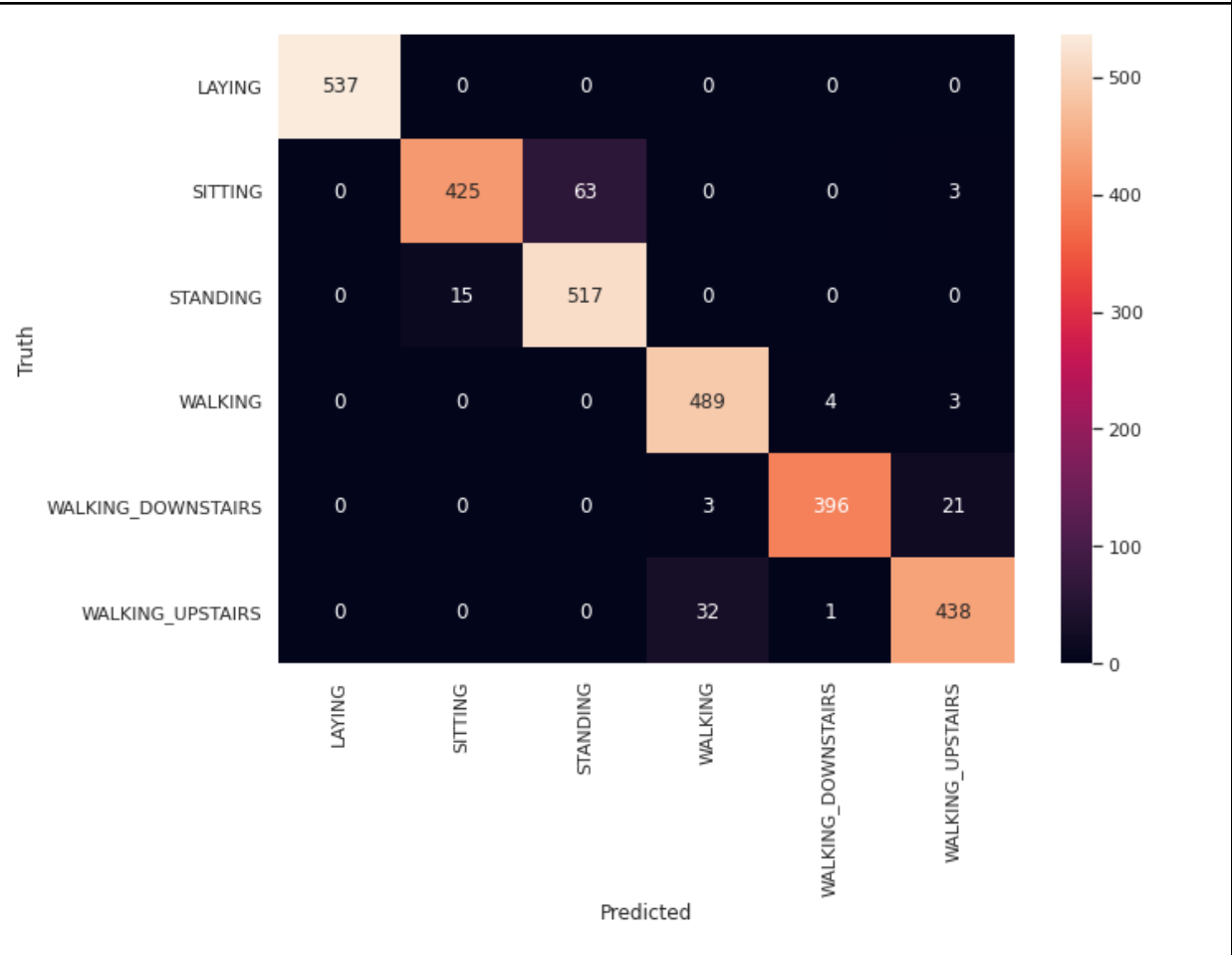| dense_32: Dense	input:	(None, 561)
	output:	(None, 100)

dense_33: Dense	input:	(None, 100)
	output:	(None, 6)
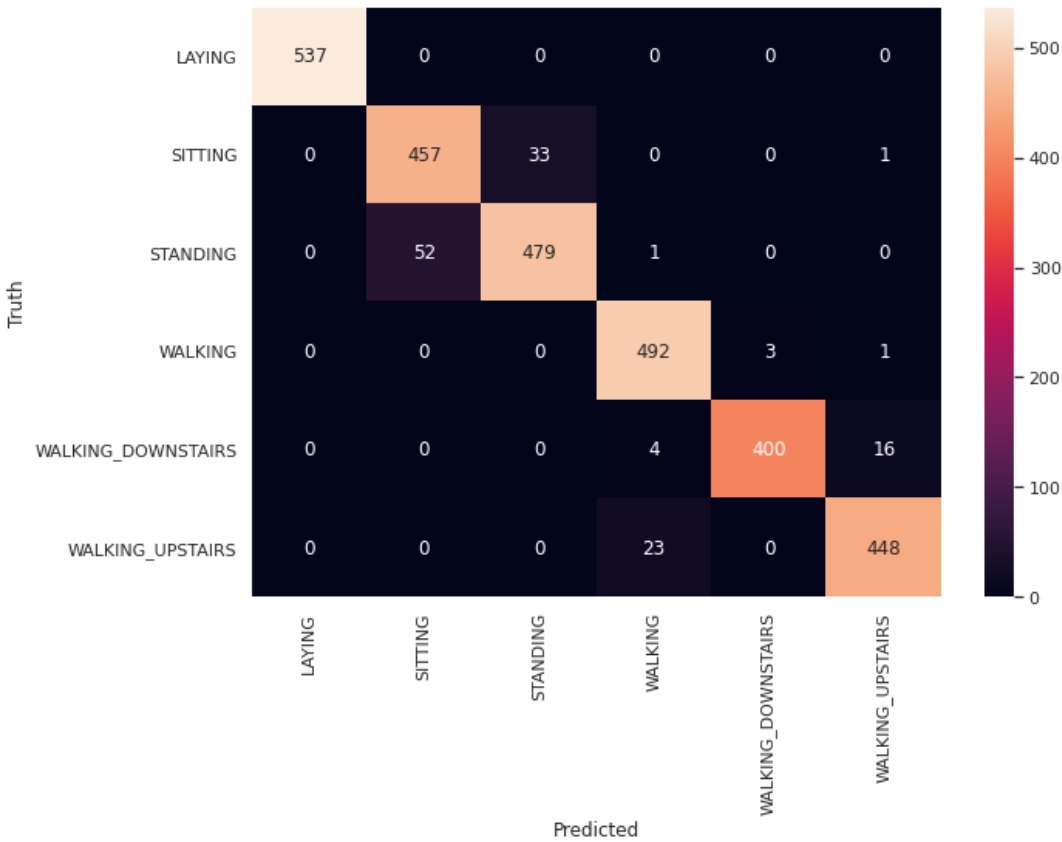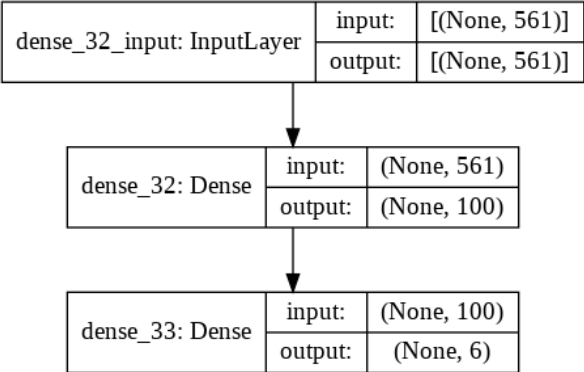
epochs = 100
Hidden layer Activation = tanh
No cross fold validation
hidden layer neurons = 100
single hidden layer

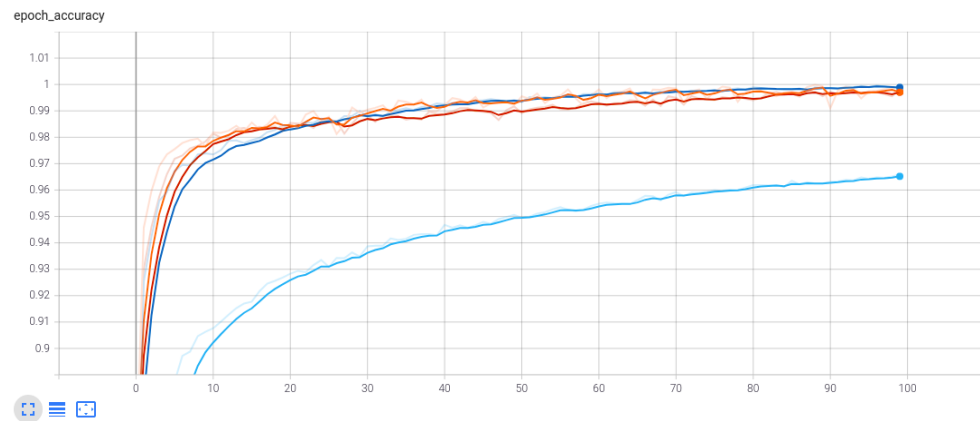# Analysing with different optimisers:

- **Adam**
- **RMSprop**
- **Adamax**
- **Adagrad**

epoch_accuracy

Fig: Comparing epoch_ accuracy with optimizers

- **Adam**
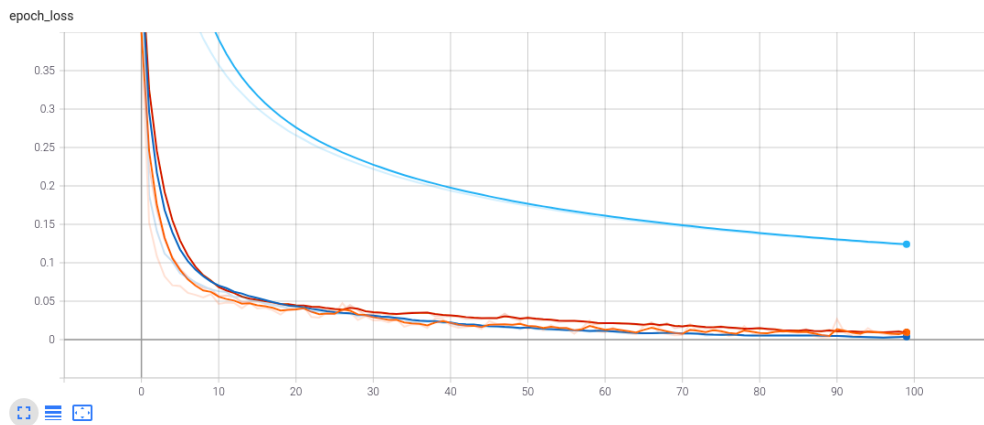- **RMSprop**
- **Adamax**
- **Adagrad**

epoch_loss

Fig: Comparing epoch_ loss with optimizers
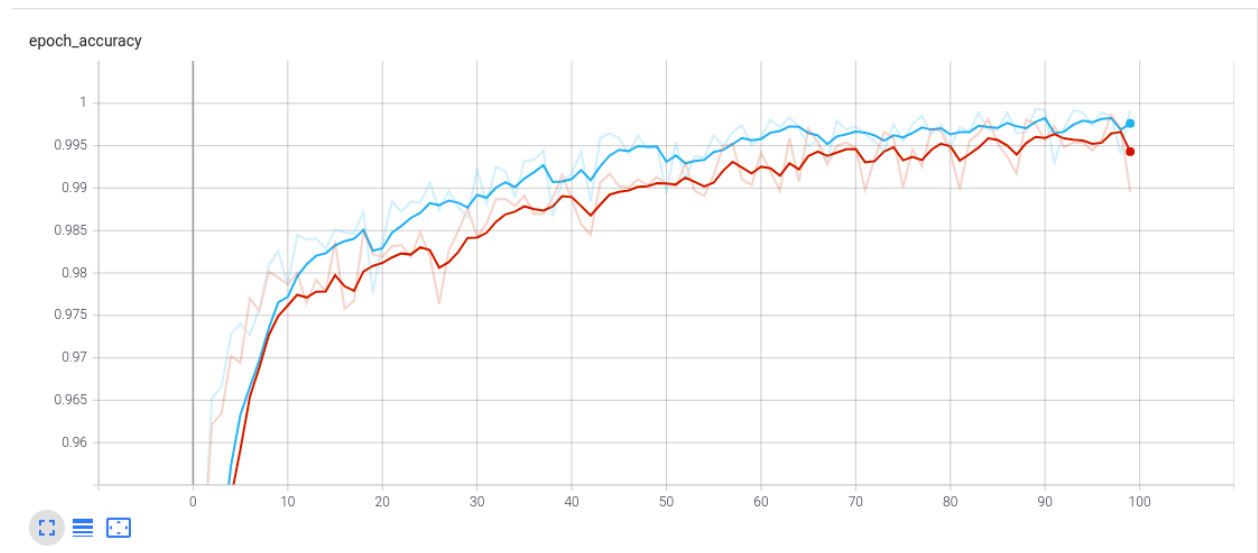
Accuracy and Loss of Model 5 and Model 6

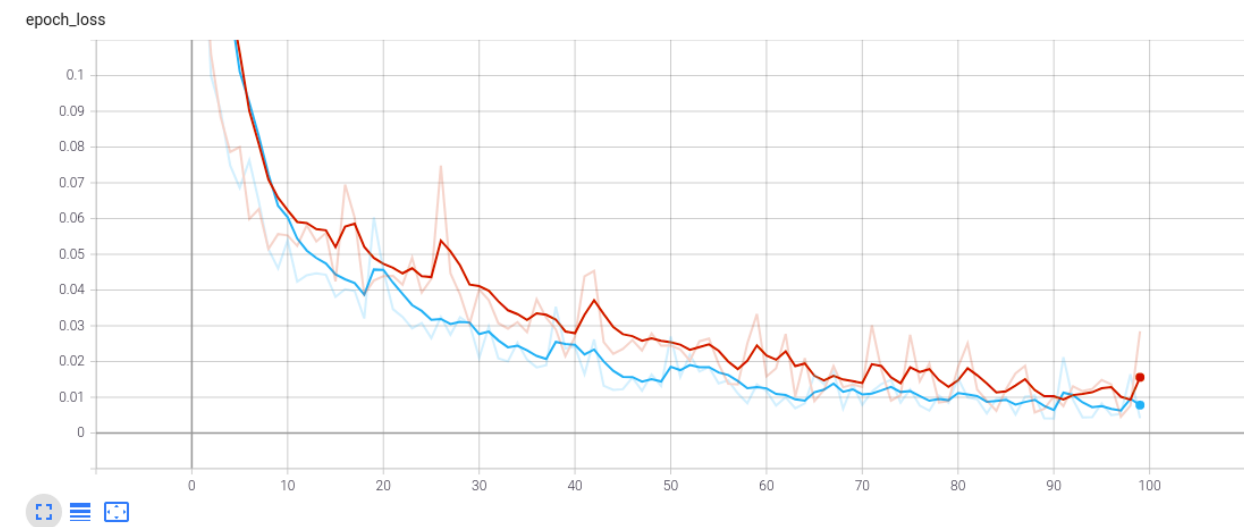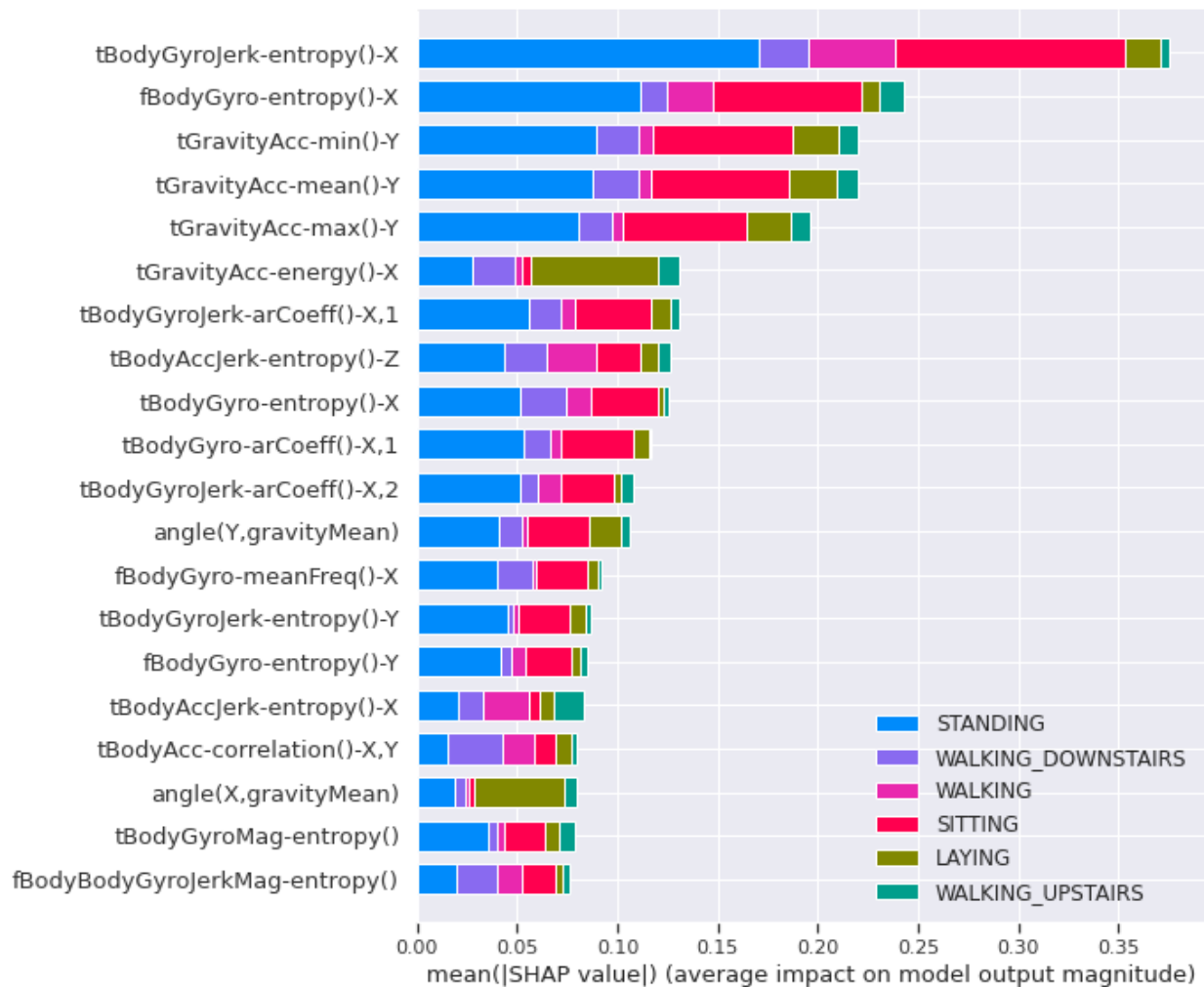Fig: Comparing epoch accuracy of Model 5 and Model 6

Fig: Comparing epoch loss for Model 5 and Model 6

Ranking of Features based on their Impact:

Observations:

1. From the Model 6 Vs Model 5 epochs and accuracy graph, we see that model 6 is performing better than model 5. Model 5 is using relu function, whereas Model 6 is using tanh function. The advantage of Tanh function is that it can center the data obtained from its previous layer.

2. From the confusion matrices, we observe that our model is getting confused between standing and sitting. It is wrongly classifying the sitting input vectors to be that of standing and standing input vectors to that of sitting. Our explanation is that in both the standing and sitting cases, the smartphone remains in a still position and hence our models find it difficult to classify the sitting and standing input vectors.

Conclusion:

We compared different neural networks learning models by changing their various hyper parameter values, for example: number of epochs, activation functions, number of hidden layers, number of hidden layer neurons. We also tried using kFoldCross Validation for testing some models.
We tried several neural networks models with different optimizers and on an average we are getting accuracy around 95% and the maximum accuracy achieved was 96.00% with Adamax optimizer.

Dataset used in this study contains data generated solely from accelerometer and gyroscope signals. This work could be improved by increasing the number of activities and situations to classify and to add data received from other sensors and devices that are commonly used in smartphones. Some of these devices are magnetometer, light sensor, proximity sensor, barometer, termometer, pedometer, heart pulse monitor, GPS and microphone. With help of these devices it would be possible to get information about the condition and location of the user and situation of the environment in order to classify much more complex activities and situations.

References:

1. E. Bulbul, A. Cetin and I. A. Dogru, "Human Activity Recognition Using Smartphones," 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 2018, pp. 1-6, doi: 10.1109/ISMSIT.2018.8567275.
2. Akram Bayat, Marc Pomplun, Duc A. Tran, A Study on Human Activity Recognition Using Accelerometer Data from Smartphones, Procedia Computer Science, Volume 34, 2014, Pages 450-457, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2014.07.009 (https://www.sciencedirect.com/science/article/pii/S1877050914008643)
3. Rasekh, Amin & Chen, Chien-An & Lu, Yan. (2014). Human Activity Recognition using Smartphone.
4. Jorge-L. Reyes-Ortiz, Luca Oneto, Albert SamÃ , Xavier Parra, Davide Anguita. Transition-Aware Human Activity Recognition Using Smartphones. Neurocomputing. Springer 2015.