Topic :Clustering-Classification Approach For Human Activity Detection Using Smartphone Dataset

Shantanu Agrawal

1901182

Course: CS360 - Machine Learning Lab

Problem Definition

Problem Definition

The objective of this project is to recognize 6 activity of a human (Sitting, Standing, Laying, Walking, Walking Downstairs, Walking Upstairs) by using gyroscope and accelerometer sensor data using smart phone

- Topics/Models Covered

- Classification
  - Logistic Regression
  - Perceptron
  - Single Layer Perceptron
  - Multi Layer Perceptron

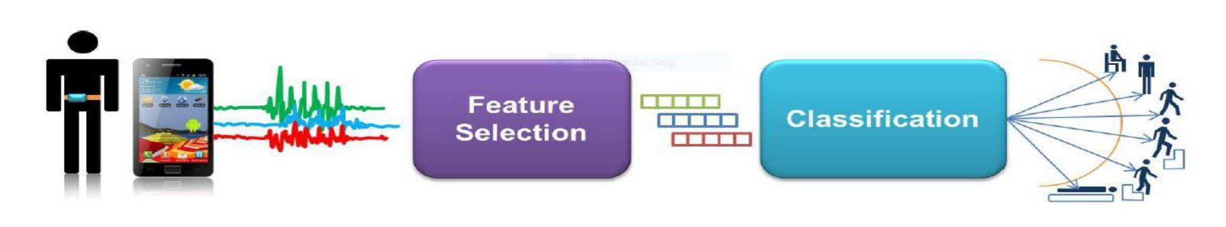
- Clustering
  - K-Means
  - K-Medoid
  - Fuzzy
  - SOM

Literature Survey

Literature Survey

➢ Paper : Human Activity Recognition on Smartphones for Mobile Context Awareness
By Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge L. Reyes-Ortiz

Setup



- Support Vector Machine(SVM) Algorithm used.
- Uses One-VS-All Approach for classification.
- Reason of using smartphone for collecting data is the high number of growth in the user in last decade and the needed sensors like gyroscope, accelerometer hardware are already present.
- In this paper, researchers are improving SVM to be used on resource limited devices by extending SVM to Hardware-Friendly SVM (HF-SVM) then to Multiclass HF-SVM (MC-HF-SVM).
- Use of MC-HF-SVM can be smartwatch since it has very small resources available.

Literature Survey

- Paper : A Public Domain Dataset for Real-Life Human Activity Recognition Using Smartphone Sensors By Daniel, Daniel, Enrique and Miguel

- The main idea is to make activity recognition use more real world data than the model data provided under UCI Human Activities Recognition (HAR) repository
- Support Vector Machine(SVM) Algorithm used due to fixed-point arithmetic data points. .
- Uses One-VS-All Approach for classification.
- Since the real world data is more unorganized, the first step is to organize it and make it easier to work with.
- More sensors are used to collect data, like GPS, Magnetometer and more types of activities recognition is aimed forUse of MC-HF-SVM can be smartwatch since it has very small resources available.
- Accuracy is slightly lower than previous works due to more real world data.

Literature Survey

- Paper : A Smartphone-Based Adaptive Recognition and Real-Time Monitoring System for Human Activities
  By Wen, Hang and Andrea

- Main idea is introduction of real-time human activity monitoring system and a adaptive learning environment.
- Unsupervised learning algorithm as we want to classify new unrecognized activity and add a label to it for future classification. This new label will be added to the class.
- Researchers are aiming for online learning algorithm which will be efficient in the terms of resources
- Hierarchical clustering has been used to learning

Result Analysis

Classification : Logistic Regression

❑ As shown in fig 1.1 by on varying epoch accuracy score

❑ Around 600 epochs convergence occur

❑ After 1000 epochs iteration

- ❖ Train Accuracy Score : 99.25%
- ❖ Test Accuracy Score : 96.13%
- ❖ Log Loss Error : 0.119389576982268
- ❖ Mean Squared Error : 0.05293518832711232

❑ Faster compare to other models

❑ Model is performing well since it is achieving 96.13% accuracy

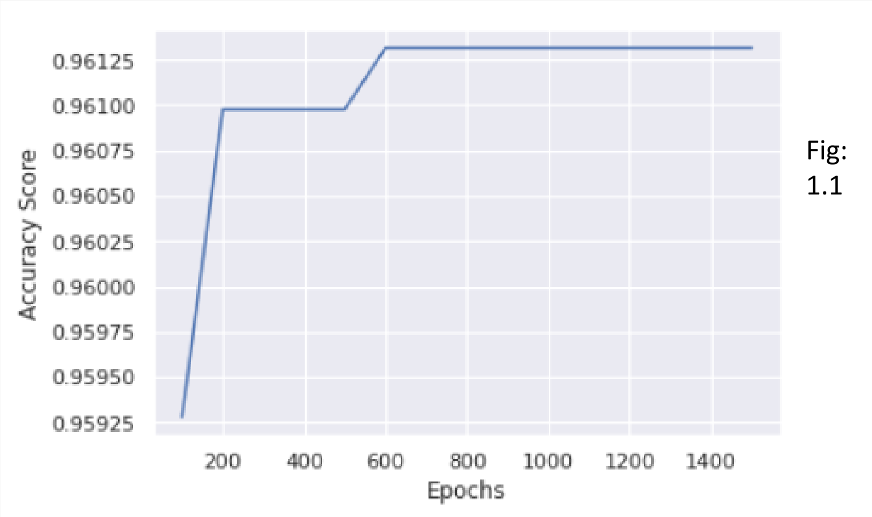
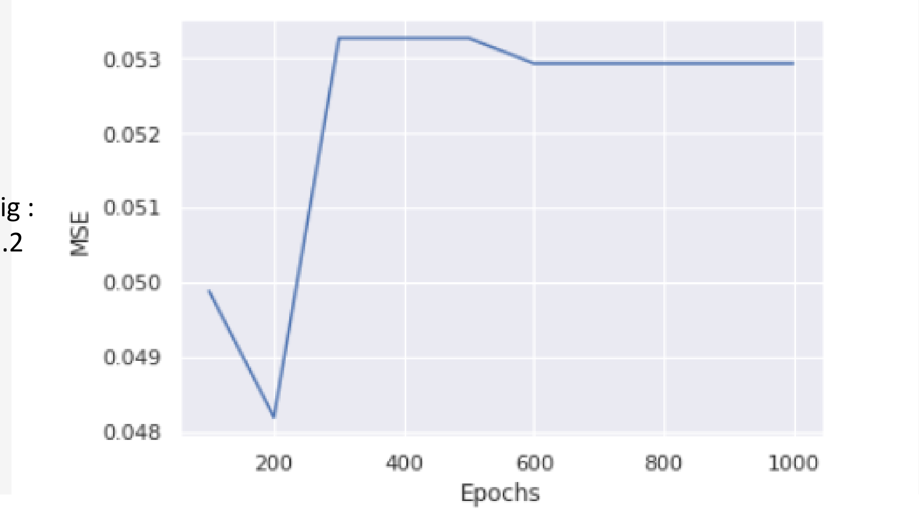


Fig: 1.1

Epoch vs Mean Squared error (Fig : 1.2)

Fig : 1.2



Classification : Stochastic Gradient Descent (ONE Vs ALL)

❑ As shown in Fig 2.1 we plot Accuracy vs epoch graph

❑ Around 800 epoch accuracy is similar

❑ After 1000 iterations

- ❖ Train Accuracy Score : 98.59%
- ❖ Test Accuracy Score : 96.13%
- ❖ Mean Squared Error : 0.04784526637258229
- ❖ Log Loss Error : 0.2194638846322902

❑ K-Fold Cross-validation is a statistical method used to estimate the skill of machine learning models.

❑ After K-Fold Validation

- ❖ Accuracy Score : 94.63%

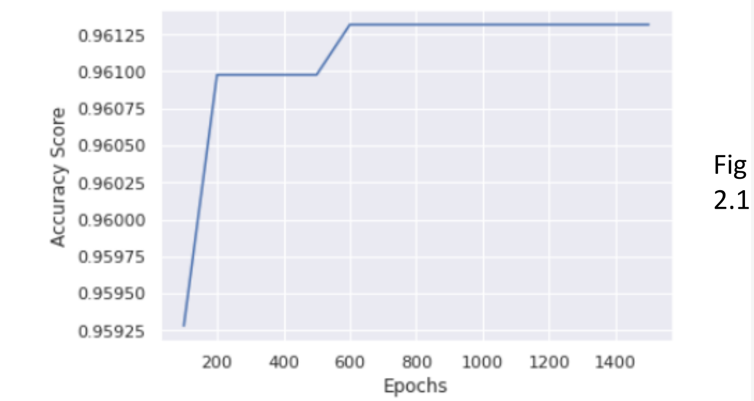


Fig : 2.1

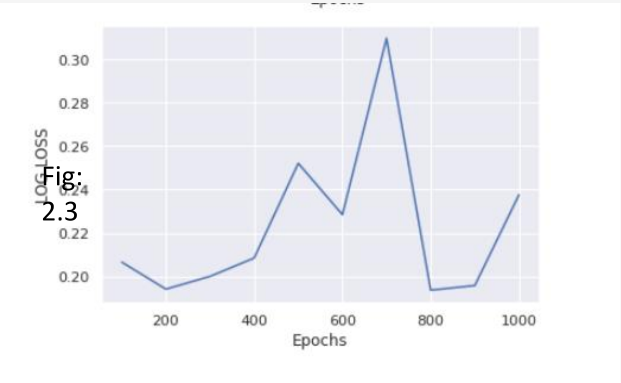
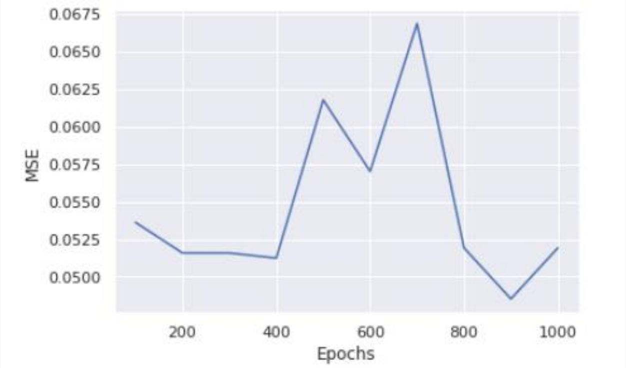


Fig: 2.3

MSE vs Epoch (fig : 2.2)

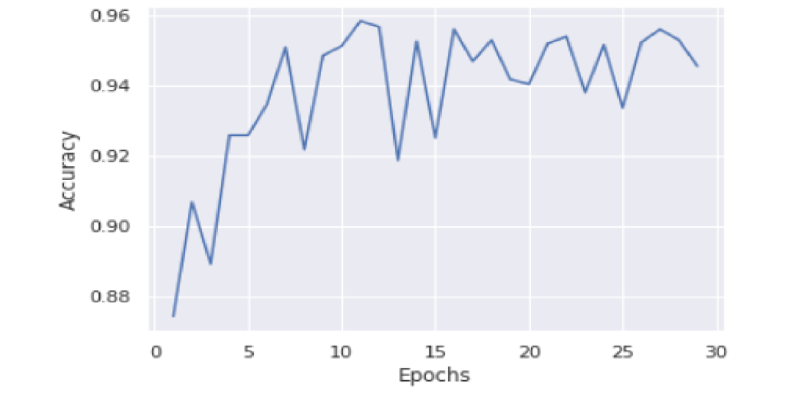


Log Loss vs Epoch (fig : 2.3)

Classification : Perceptron

- ❑ It does not require a learning rate
- ❑ It updates its model only on mistakes.

Fig: 3.1



❑As shown in fig 3.1 by on varying epoch accuracy score

❑After 30 epochs iteration

- ❖ Train Accuracy Score : 95.06%
- ❖ Test Accuracy Score : 90.73%
- ❖ `Mean Squared Error : 0.10960298608754666`

❑ In MSE vs Epoch (fig : 3.2)
After 20 epoch MSE is almost constant

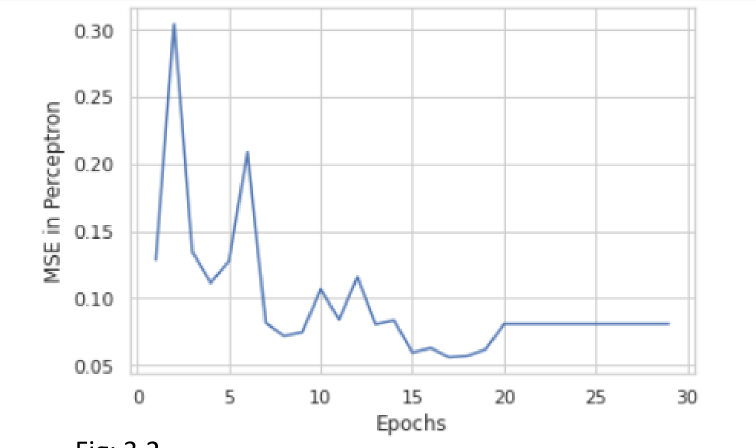
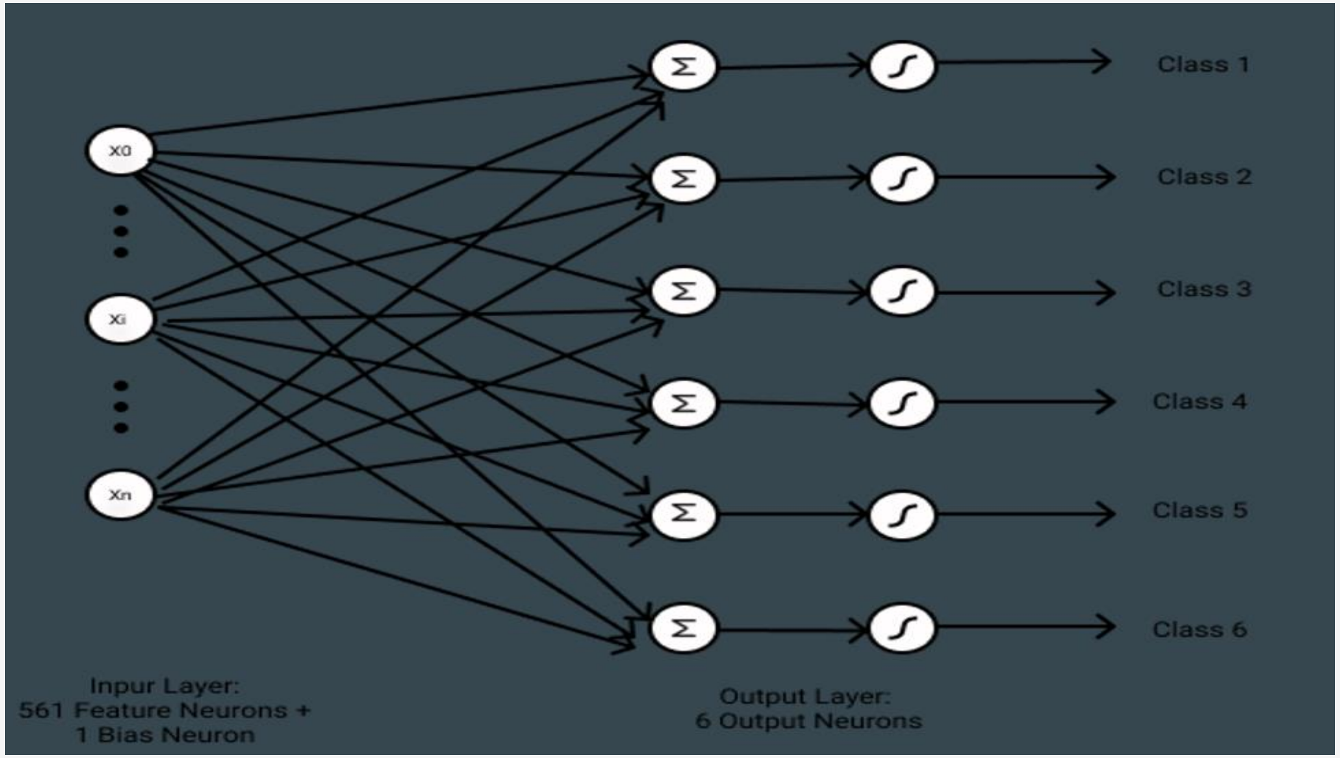


Fig: 3.2

Classification : Single Layer Perceptron (SLP)

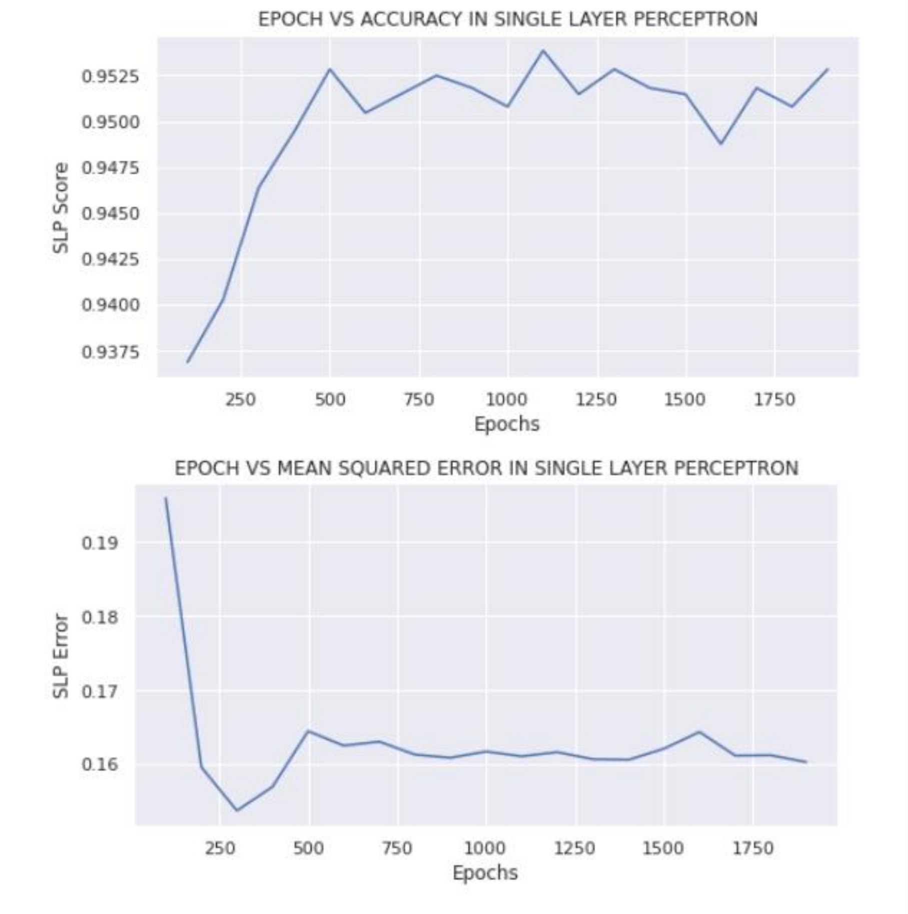
- ❑ It is a feed-forward neural network
- ❑ It has a single layer of perceptrons or artificial neurons. It updates its model only on mistakes.

Architecture



Classification : Single Layer Perceptron (SLP)

- □ Slower compared to other models
- □ Sigmoid activation function is used
- □Model converges with high accuracy
- □Higher accuracy achieved 95.35%



Classification : Multi Layer Perceptron (MLP)

❑It has one input layer, one output layer, and multiple hidden layers (feed-forward architecture).

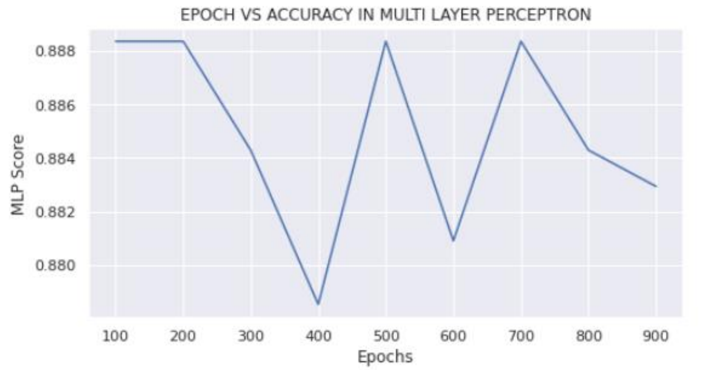
❑No of hidden neuron taken is 562

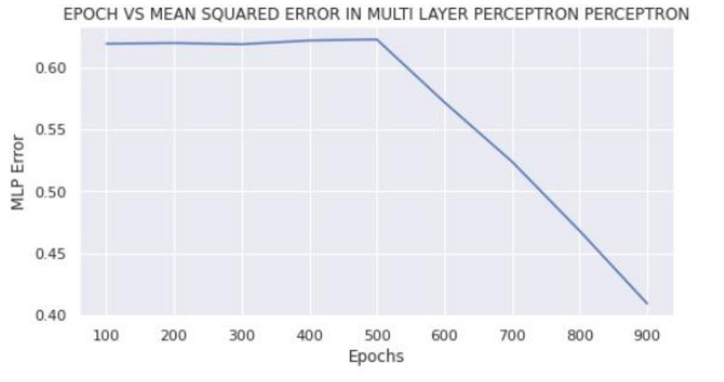
❑Slow to train for increased number of layers and nodes

❑Slow to predict in case of large number of layers

❑Not suitable for real time accurate computations

❑Highest accuracy achieved 88.89%





Clustering

- ❑Fast compared to other models.
- ❑Algorithms Used – K-Means, K-Medoids, Fuzzy Clustering, SOM
- ❑Model converges with very low accuracy
- ❑Highest Accuracy achieved in K-Means – 32.37%
- ❑Highest Accuracy achieved in K-Medoid – 20.0543%
- ❑Highest Accuracy achieved in Fuzzy – 24.8728%
- ❑Highest Accuracy achieved in SOM – 9.3482%
- ❑Poor accuracy even with the reduced features.

Refernces

- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Perceptron.html
- https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html
- https://scikit-learn-extra.readthedocs.io/en/stable/generated/sklearn_extra.cluster.KMedoids.html
- https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- https://pypi.org/project/fuzzy-c-means/