Why MultiLayer Perceptron/Neural Network?

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.
Other advantages include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
2. One of the preferred techniques for gesture recognition.
3. MLP/Neural networks do not make any assumption regarding the underlying probability density functions or other probabilistic information about the pattern classes under consideration in comparison to other probability based models [1].
4. They yield the required decision function directly via training.
5. A two layer backpropagation network with sufficient hidden nodes has been proven to be a universal approximator [2] [3].

Objective:

The Neural network was implemented to recognize the three hand gestures namely grasp, point and push, irrespective of who is doing these hand gestures.

Attributes:

Attribute Extraction:

- The first step was to extract relevant attributes from the data. Following attributes were extracted for neural nets:
    1. Sumdis = Sum of the distances of 3 finger markers from the centroid, figure 2.
    2. SumdeltaX = Sum of the differences of X displacement of each finger marker with respect to the X displacement of centroid, figure 1.
    3. SumdeltaY = Sum of the differences of Y displacement of each finger marker with respect to the Y displacement of centroid, figure 1.
    4. SumdeltaZ = Sum of the differences of Z displacement of each finger marker with respect to the Z displacement of centroid, figure 1.

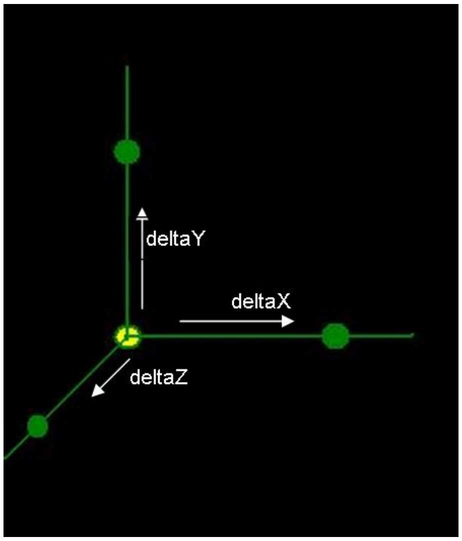


Figure1 **Figure 2**

Extracted Features for MLP

Attribute Selection:

- Next step was attribute selection. Initially Principal Component Analysis was used for attribute selection, but a poor discrimination between gestures was observed when the data was projected on the PCA axes. Below are the results from PCA and the figure depicting the projections on PCA axes (figures were generated using WEKA, which does not have feature to save figures, hence screen shots are provided):-

Instances: 3821
Attributes: 5
Label
SumdeltaX
SumdeltaY
SumdeltaZ
Sumdis
=== Attribute Selection on all input data ===
Correlation matrix

1	-0.07	0.57	0.22
-0.07	1	0.54	-0.71
0.57	0.54	1	-0.17
0.22	-0.71	-0.17	1

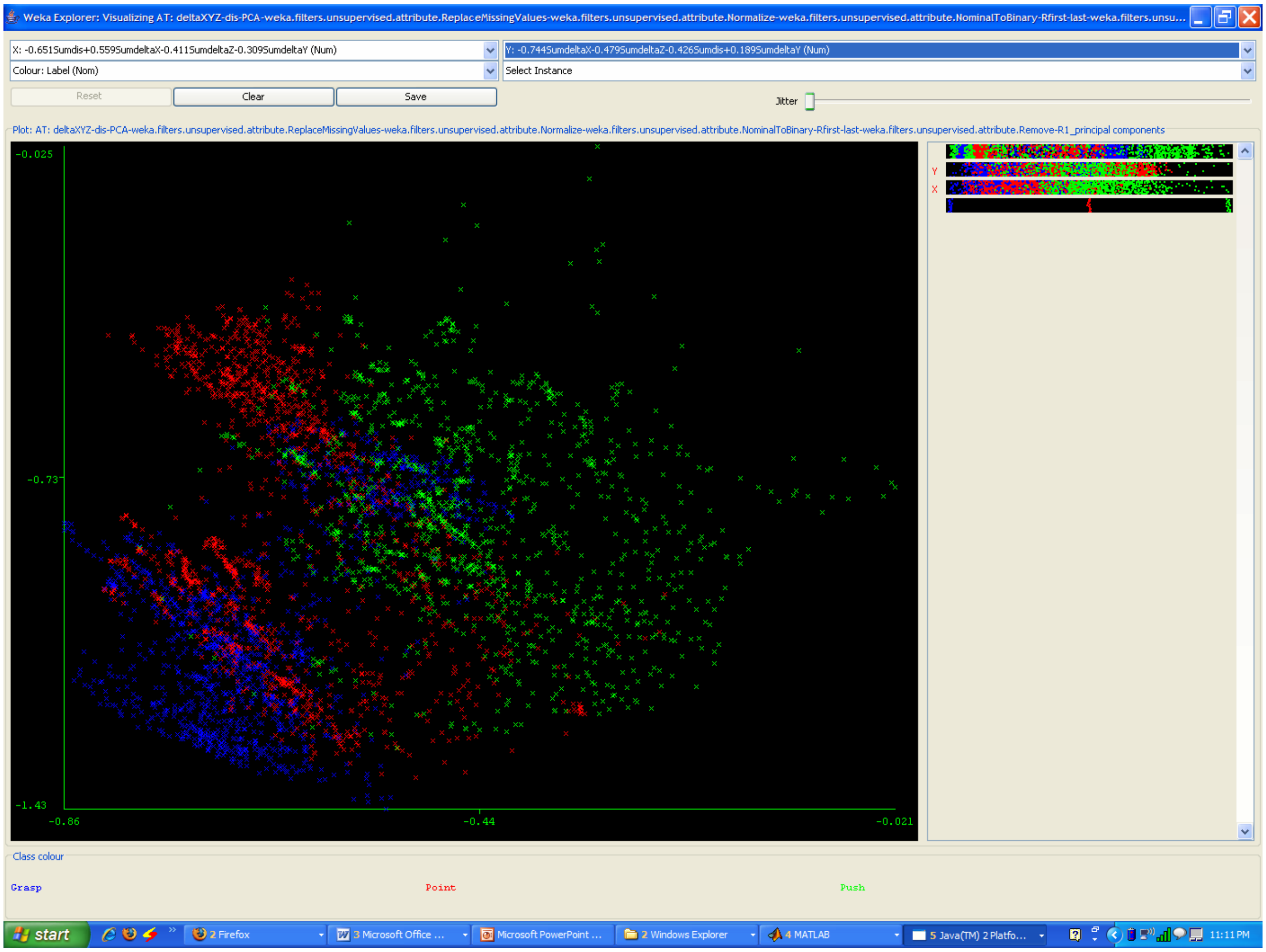
eigenvalue	proportion	cumulative	
1.9907	0.49768	0.49768	-0.654SumdeltaY+0.529Sumdis-0.525SumdeltaZ-0.131SumdeltaX
1.51151	0.37788	0.87555	-0.744SumdeltaX-0.479SumdeltaZ-0.426Sumdis+0.189SumdeltaY
0.3634	0.09085	0.9664	-0.651Sumdis+0.559SumdeltaX-0.411SumdeltaZ-0.309SumdeltaY

Eigenvectors

V1	V2	V3	
-0.1309	-0.744	0.5586	SumdeltaX
-0.6543	0.189	-0.3087	SumdeltaY
-0.5246	-0.4791	-0.4107	SumdeltaZ
0.5287	-0.4257	-0.6512	Sumdis

Ranked attributes:
0.5023 1 -0.654SumdeltaY+0.529Sumdis-0.525SumdeltaZ-0.131SumdeltaX
0.1244 2 -0.744SumdeltaX-0.479SumdeltaZ-0.426Sumdis+0.189SumdeltaY
0.0336 3 -0.651Sumdis+0.559SumdeltaX-0.411SumdeltaZ-0.309SumdeltaY

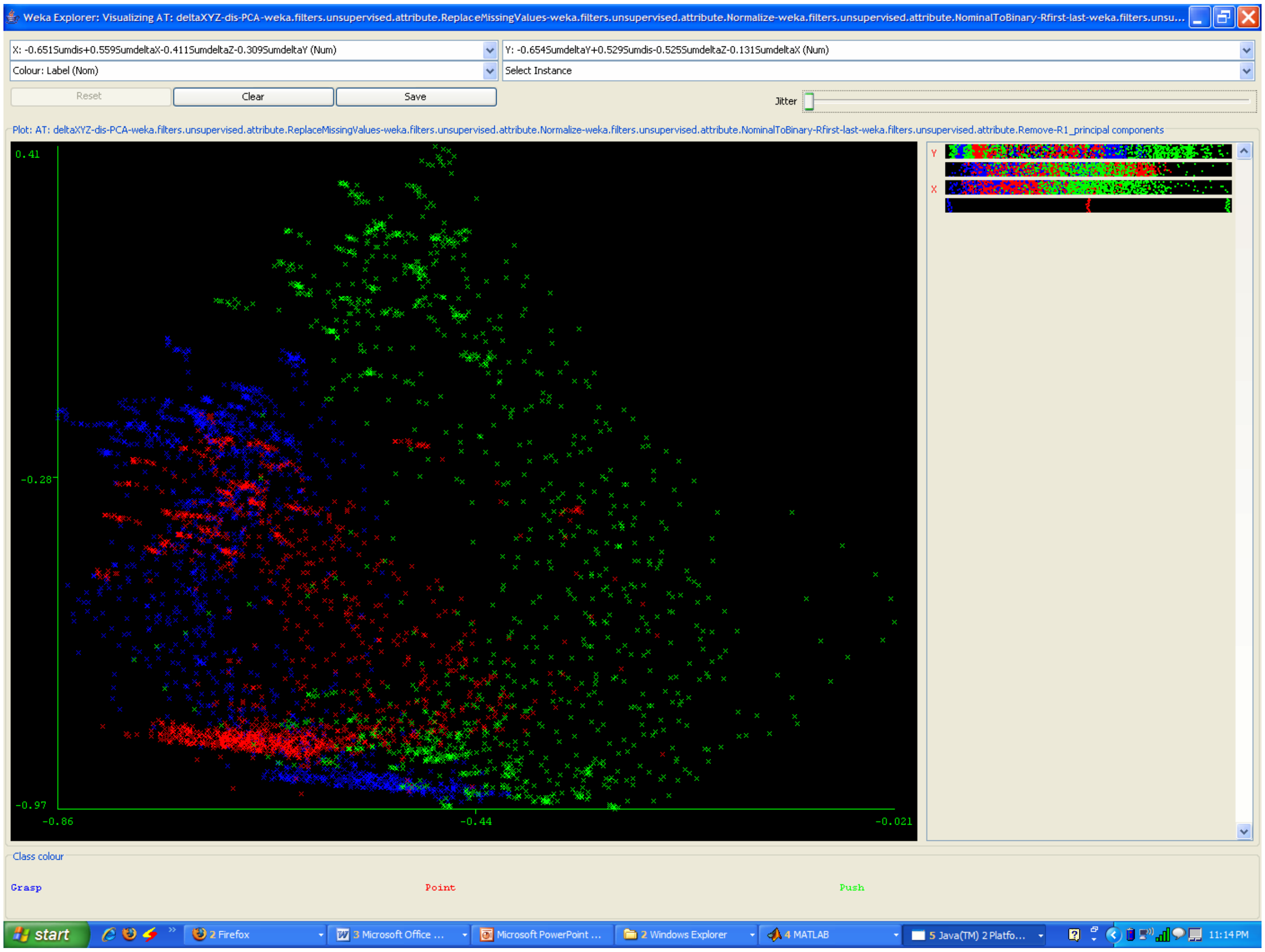
Selected attributes: 1,2,3 : 3



X aixs -0.651Sumdis+0.559SumdeltaX-0.411SumdeltaZ-0.309SumdeltaY

Y axis 0.1244 2 -0.744SumdeltaX-0.479SumdeltaZ-0.426Sumdis+0.189SumdeltaY

Grasp Point Push



X axis -0.651Sumdis+0.559SumdeltaX-0.411SumdeltaZ-0.309SumdeltaY
Y axis -0.654SumdeltaY+0.529Sumdis-0.525SumdeltaZ-0.131SumdeltaX
Grasp Point Push

Therefore to decide upon the attributes heuristics, observations and domain knowledge were used and three attributes namely Sumdis, SumdeltaY and SumdeltaZ were selected.

Models:

From the selected features two models were created one with 2 dimensional feature space (Sumdis and SumdeltaY) and a 3 dimensional feature space (Sumdis, SumdeltaY and SumdeltaZ). Two models were created to observe the effect of dimensionality (increasing the number of features) and also after the classification results from 2D features space model it was observed that increasing a feature i.e. SumdeltaZ (which was logical as there is a major difference between the displacements along Z, of Grasp and Point gestures). All the inputs to the neural net were normalized in the range of -1 to 1.

Model paramenters:-

- Parameters for MLP using 2D feature space (figure 3)
  - ❑ No. of Hidden layer 1 with 2 hidden nodes,
  - ❑ Learning rate = 0.3, Momentum = 0.2,
  - ❑ Epochs =600, sigmoid for activation.

Attributes: 2
Sumdis
SumdeltaY

=== Classifier model (full training set) ===

Sigmoid Node 0
Inputs   Weights
Threshold   -2.873084657868268
Node 3   11.098430581589703
Node 4   1.6427091491248684
Sigmoid Node 1
Inputs   Weights
Threshold   -20.95588207113826
Node 3   -15.325529796449096
Node 4   22.676419665124076
Sigmoid Node 2
Inputs   Weights
Threshold   4.462354683446656
Node 3   -6.567587151859792
Node 4   -7.1281632679527664
Sigmoid Node 3
Inputs   Weights
Threshold   -64.76248623634478
Attrib Sumdis   71.06468600599895
Attrib SumdeltaY   9.684557178979185
Sigmoid Node 4
Inputs   Weights
Threshold   -38.58035032176554
Attrib Sumdis   53.054453951951984
Attrib SumdeltaY   12.241053571601267
Class Gesture-Grasp
Input
Node 0
Class Gesture-Point
Input
Node 1
Class Gesture-Push
Input
Node 2

- Parameters for MLP using 3D feature space (figure 4)
  - ❑ No. of Hidden layer 1 with 3 hidden nodes,
  - ❑ Learning rate = .12, Momentum = 0.2,
  - ❑ Epochs =600, sigmoid for activation.

Attributes: 3
Sumdis
SumdeltaY
SumdeltaZ

=== Classifier model (full training set) ===

Sigmoid Node 0
Inputs   Weights
Threshold   5.830553233849687
Node 3   23.316296519360424
Node 4   -6.451972450033389
Node 5   -25.683060546064823
Sigmoid Node 1
Inputs   Weights
Threshold   -3.854105033173585
Node 3   -16.492886626269794
Node 4   -15.001763796835775

Node 5 16.32162785615188
Sigmoid Node 2
Inputs Weights
Threshold -3.9686026084417025
Node 3 2.1228565627305205
Node 4 7.189881343808142
Node 5 -0.2448461289340373
Sigmoid Node 3
Inputs Weights
Threshold -21.219979824375333
Attrib Sumdis -4.715087227400818
Attrib SumdeltaY 9.775137470000569
Attrib SumdeltaZ 23.380821573194535
Sigmoid Node 4
Inputs Weights
Threshold 3.373396889194885
Attrib Sumdis -15.147882030310672
Attrib SumdeltaY -14.371480659786206
Attrib SumdeltaZ -2.2276558260363553
Sigmoid Node 5
Inputs Weights
Threshold 7.2777490855067235
Attrib Sumdis -19.253172825505754
Attrib SumdeltaY -8.256028387585946
Attrib SumdeltaZ 6.636471959817268
Class Gesture-Grasp
Input
Node 0
Class Gesture-Point
Input
Node 1
Class Gesture-Push
Input
Node 2

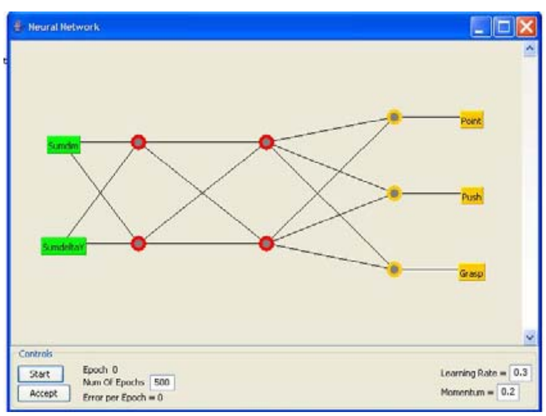


Figure 3

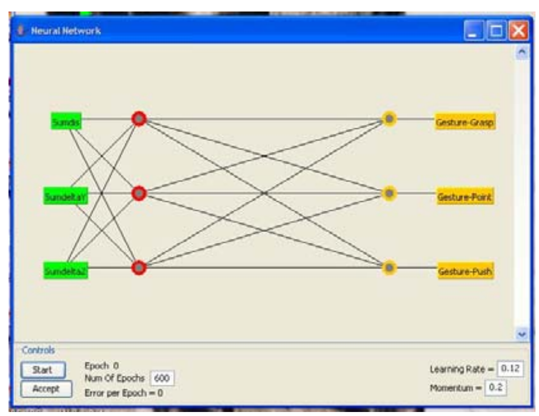


Figure 4

Neural Network for 2D & 3D feature space

The above mentioned parameters were derived after experimenting with various parameters, using these the best classification was achieved.

Results:

- 2D Feature space
  - ❑ **10 folds cross validation on Manu's Data**

- Correctly Classified Instances 3171 80.1567 %
- Incorrectly Classified Instances 785 19.8433 %
- Root mean squared error 0 .3058
- Kappa statistic 0.7023
- Mean absolute error 0.1929
- Relative absolute error 43.4131 %
- Root relative squared error 64.8791 %
- Total Number of Instances 3956

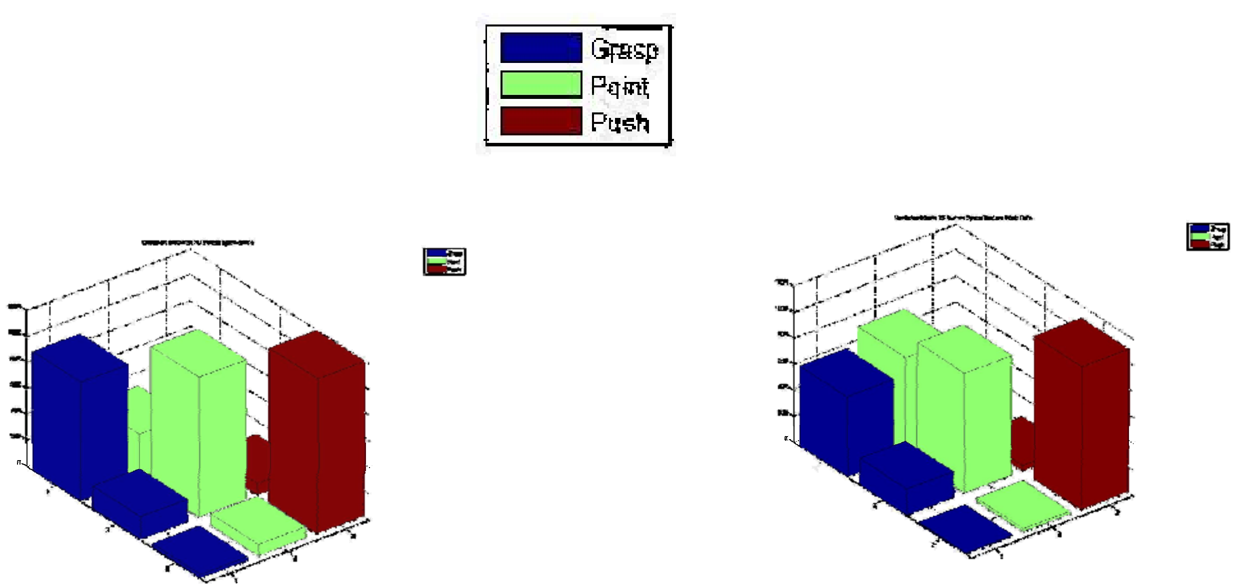
=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.689	0.076	0.82	0.689	0.749	Gesture-Grasp
0.804	0.161	0.718	0.804	0.759	Gesture-Point
0.914	0.062	0.879	0.914	0.896	Gesture-Push

- **Model trained on Manu's data and tested on Rita's data**
- Correctly Classified Instances 2622 71.9144 %
- Incorrectly Classified Instances 1024 28.0856 %
- Kappa statistic 0.5825
- Mean absolute error 0.2159
- Root mean squared error 0.3628
- Relative absolute error 48.5867 %
- Root relative squared error 76.9631 %
- Total Number of Instances 3646

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.447	0.09	0.748	0.447	0.559	Gesture-Grasp
0.792	0.307	0.547	0.792	0.647	Gesture-Point
0.977	0.023	0.95	0.977	0.963	Gesture-Push



2D feature space -Manu (10Folds X validation) **2D feature space test on Rita's data**

Confusion Matrix

a b c <-- classified as	a b c <-- classified as
910 336 75 | a = Gesture-Grasp	610 741 15 | a = Gesture-Grasp
173 1073 89 | b = Gesture-Point	200 919 42 | b = Gesture-Point
27 85 1188 | c = Gesture-Push	5 21 1093 | c = Gesture-Push

- 3D Feature space
  - ❑ **10 folds cross validation on Manu's Data**
  - ❑ Correctly Classified Instances 3511 91.8869 %
  - ❑ Incorrectly Classified Instances 310 8.1131 %
  - ❑ Kappa statistic 0.8779
  - ❑ Mean absolute error 0.0916
  - ❑ Root mean squared error 0.2112
  - ❑ Relative absolute error 20.6521 %
  - ❑ Root relative squared error 44.8404 %
  - ❑ Total Number of Instances 3821

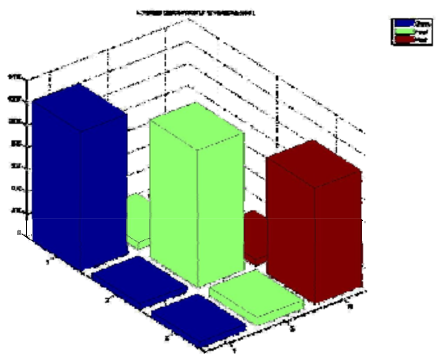
=== Detailed Accuracy By Class ===

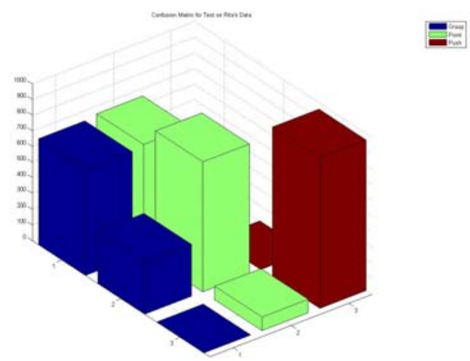
TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.945	0.043	0.92	0.945	0.932	Gesture-Grasp
0.92	0.055	0.9	0.92	0.91	Gesture-Point
0.888	0.024	0.941	0.888	0.914	Gesture-Push

  - ❑ **Model trained on Manu's data and tested on Rita's data**
  - ❑ Correctly Classified Instances 2463 68.953 %
  - ❑ Incorrectly Classified Instances 1109 31.047 %
  - ❑ Kappa statistic 0.5349
  - ❑ Mean absolute error 0.2277
  - ❑ Root mean squared error 0.4343
  - ❑ Relative absolute error 51.3569 %
  - ❑ Root relative squared error 92.2682 %
  - ❑ Total Number of Instances 3572

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.493	0.15	0.67	0.493	0.568	Gesture-Grasp
0.714	0.322	0.516	0.714	0.599	Gesture-Point
0.919	0	1	0.919	0.958	Gesture-Push



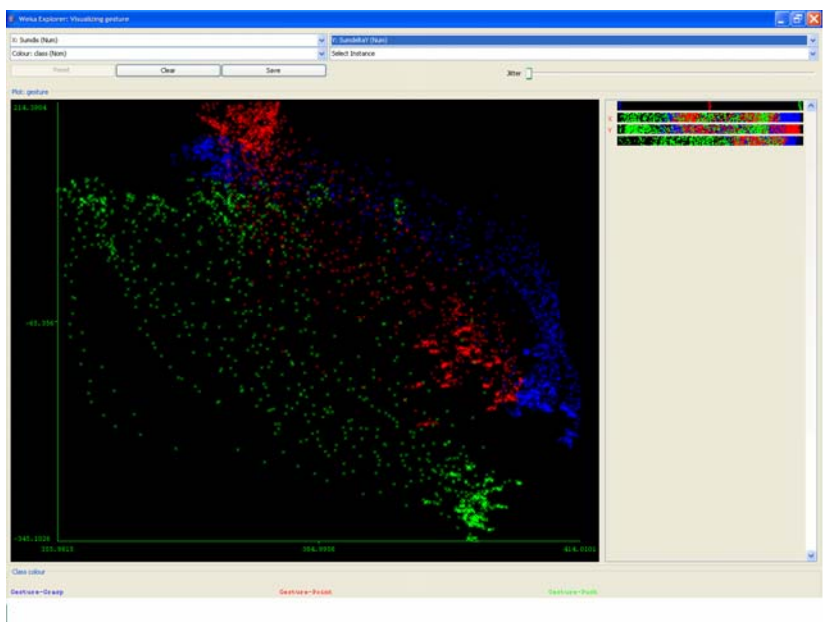


3D feature space -Manu (10Folds X validation) **3D feature space test on Rita's data**

Confusion Matrix

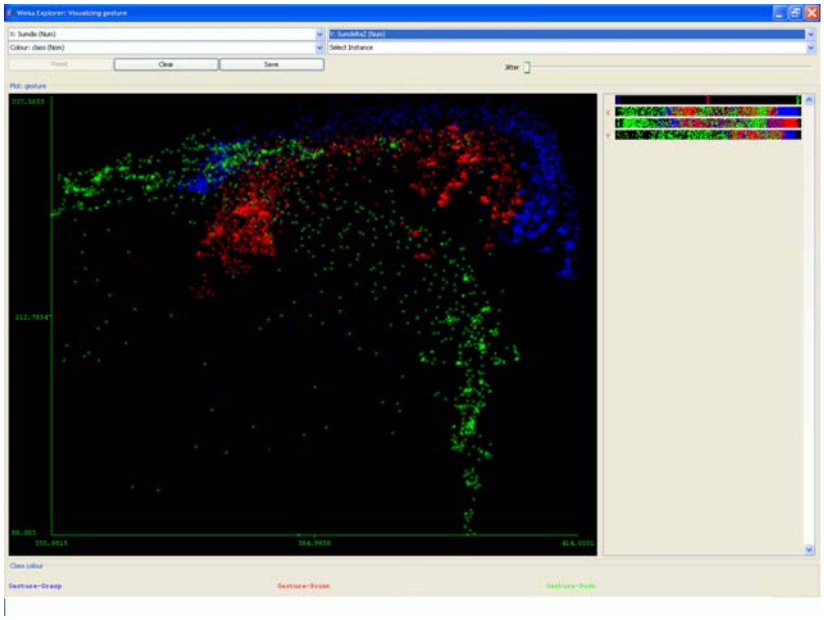
a	b	c	<-- classified as	a	b	c	<-- classified as
1248	62	11 |	a = Gesture-Grasp	674	692	0 |	a = Gesture-Grasp
53	1228	54 |	b = Gesture-Point	332	829	0 |	b = Gesture-Point
55	75	1035 |	c = Gesture-Push	0	85	960 |	c = Gesture-Push

Feature space plots:-

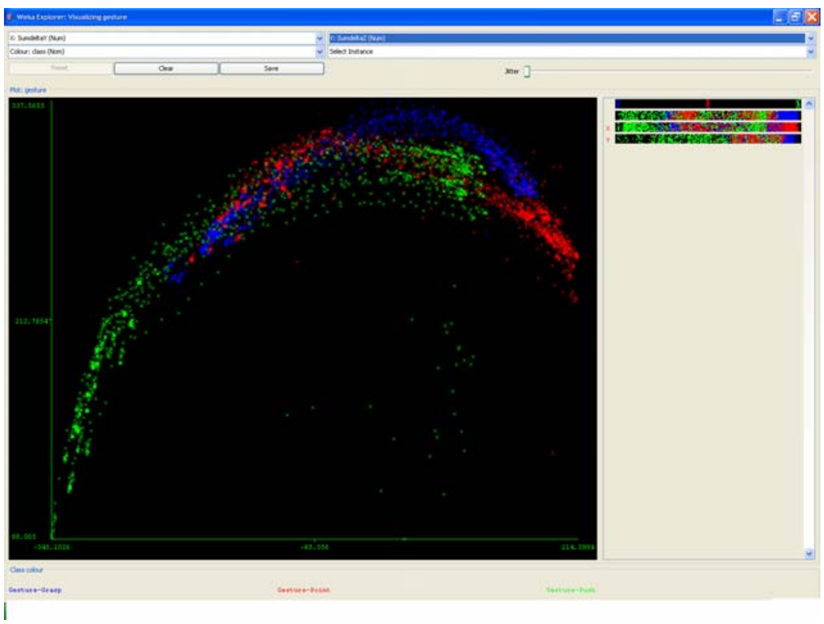


X axis - Sumdis Vs Y axis - SumdeltaY

Grasp Point Push



X axis - Sumdis Vs Y axis - SumdeltaZ
Grasp Point Push



X axis - SumdeltaY Vs Y axis - SumdeltaZ
Grasp Point Push

Conclusions: -

- Adding extra feature i.e. increasing dimensionality does not help in this case.
- In comparison to 2 features, though good results were observed for 10 folds X validation, but the performance degraded for test data (look like overfitting).
- For 3 features more point gestures were misclassified as grasp, but more grasp gestures were misclassified as point for 2 features. A tradeoff between increasing the total classification accuracy and true positives can be seen, the neural net was not able to optimize this.
- In both cases gesture Push was unambiguously recognized with True positive rate as high as 0.977.

- For MLP deciding upon learning rate is very important, a lower rate performed better in 3D feature spaces.
- After experimenting with different number of hidden layer and hidden node, it was found that a single hidden layer with few hidden nodes performed better. Adding extra hidden layer does not help always, but increasing the number of nodes might help.

Discussion & future work: -

Their ability to learn by example makes neural nets very flexible and powerful. There is no need to devise an algorithm in order to perform a specific task; i.e. there is no need to understand the internal mechanisms of that task. Along various other advantages of Neural nets there disadvantages too they cannot be programmed to perform a specific task; the examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. Also, network finds out how to solve the problem by itself, hence its operation can be unpredictable. The problem with the backpropagation algorithm is that it tries to find a local minimum in the error function output, if it ends up in finding the wrong one, the results can drastically bad, and that's why learning rate is important. Instead of using a simple backpropagation algorithm advanced algorithms like hyper rectangular composite NN (HRCNN) using supervised decision directed learning (SDDL) can be used [1]. More appropriate features can be extracted. And a well modeled neural net can be developed for real time hand gesture recognition.

Reference:

1. M. C. Su, W. F. Jean, and H. T. Chang, 1996, " A Static Hand Gesture Recognition System Using a Composite Neural Network," in Fifth IEEE Int. Conf. on Fuzzy Systems, pp. 786-792, New Orleans, U.S.A. (NSC85-2213-E-032-009)
2. G. Cybenko, "Approximation by superpositions of a sigmoidal func-. tion," Math. Contr., Signals, Syst., vol. 2, pp. 303–314, 1989
3. K. Hornik, M. Stinchcombe and H. White (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2, 359-366.