Car Evaluation

Shivam Porwal(1611CS14)

Debanjan Sarkar(1611CS17)

Background and Motivation

- We all Like Cars.
- Car Sell:
  - Around 3,50,000 cars are sold in India every year.
  - 5,15,000 cars are sold in Taiwan, 2005(which was highest in 10 years).

Introduction of Dataset

- Car Evaluation Database:
  - UCI Machine Learning Depository
  - https://archive.ics.uci.edu/ml/datasets/Car+Evaluation.
- Model Evaluates the cars based on certain characteristics.
- Derived from a Simple Hierarchical decision model.
- It is a Multiclass Classification Problem.

How Much Data is Enough ?..

How Much Data is Enough ?...

- We try to Discover the useful knowledge from volumes of data.

How Much Data is Enough ?...

- We try to Discover the useful knowledge from volumes of data.
- It Means larger data we use, better result we can get.

How Much Data is Enough ?...

- We try to Discover the useful knowledge from volumes of data.
- It Means larger data we use, better result we can get.
- But some scholar think that great deal of data don't guarantee better result than little of data.

How Much Data is Enough ?...

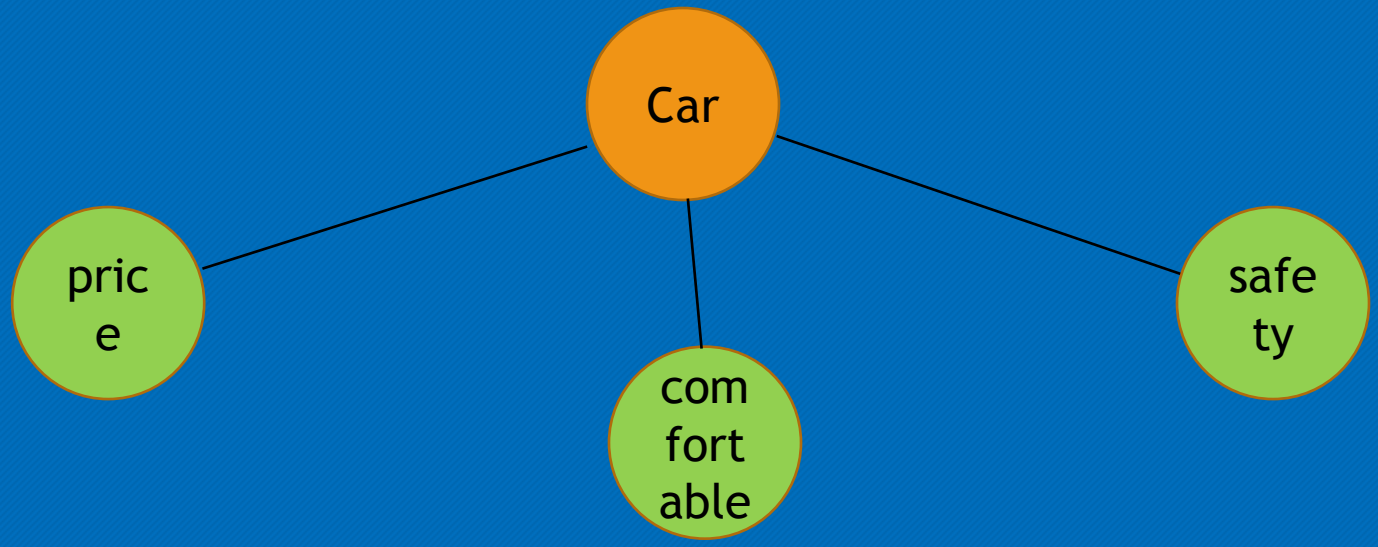
- We try to Discover the useful knowledge from volumes of data.
- It Means larger data we use, better result we can get.
- But some scholar think that great deal of data don't guarantee better result than little of data.
- Because resources are limited, the large samples will result in much load and contain lots of exceptional cases.

DataSet Information

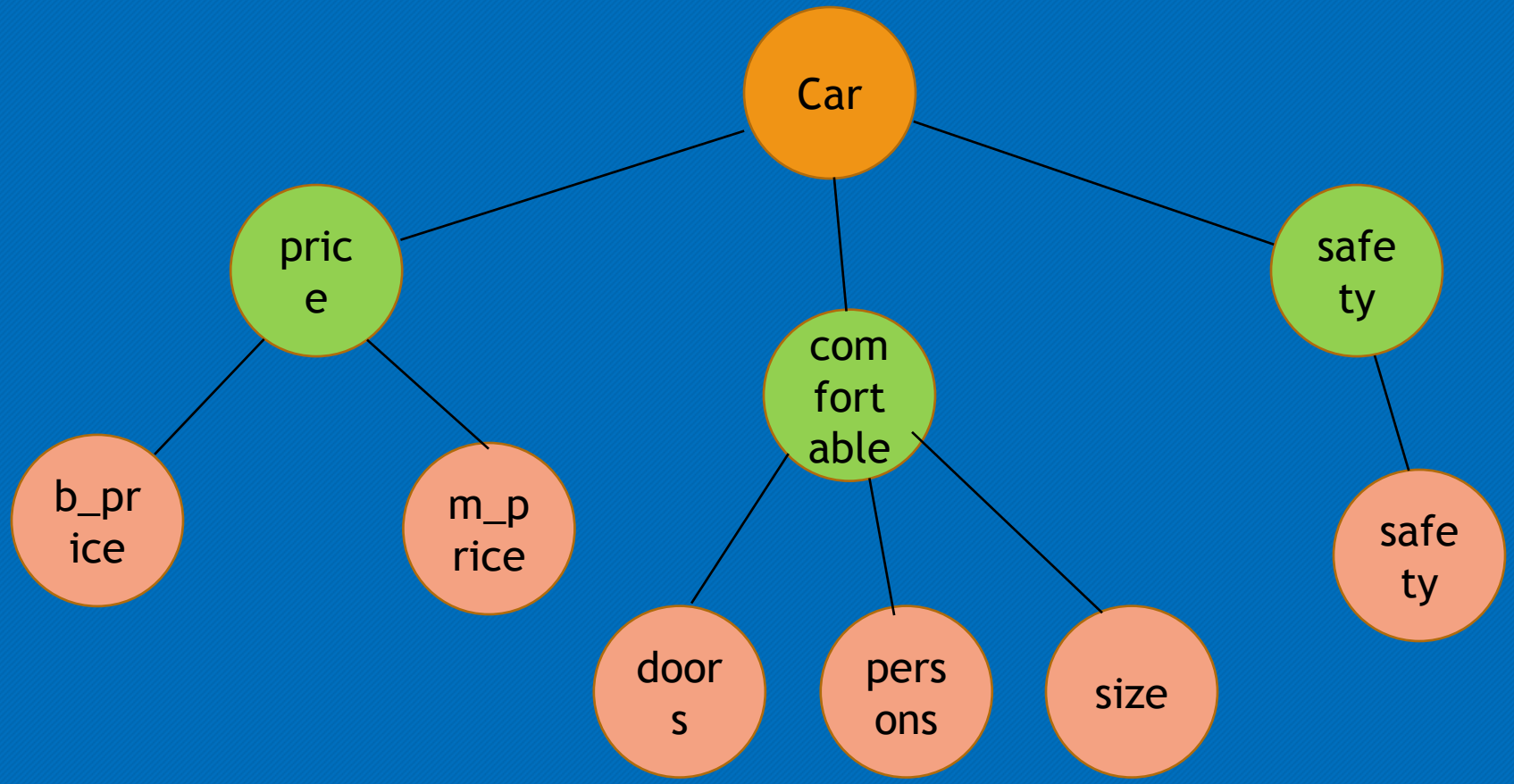
- Dataset consist of 1728 instances.
- Each record consist of 6 attributes:
  - Buying {very high, high, med, low}
  - MAINT {very high, high, med, low}
  - Doors {2,3,4,5-more}
  - Person {2,4,more}
  - LUG_BOOT {small, med, big}
  - Safety {low, Med, High}
- Class {unacc, acc, good, very-good}

Know the data

Know the data



Know the data



Approach

- Solution with neural network.
- As we cannot work with strings in neural networks, so we map the attribute values and classes to binary numbers.

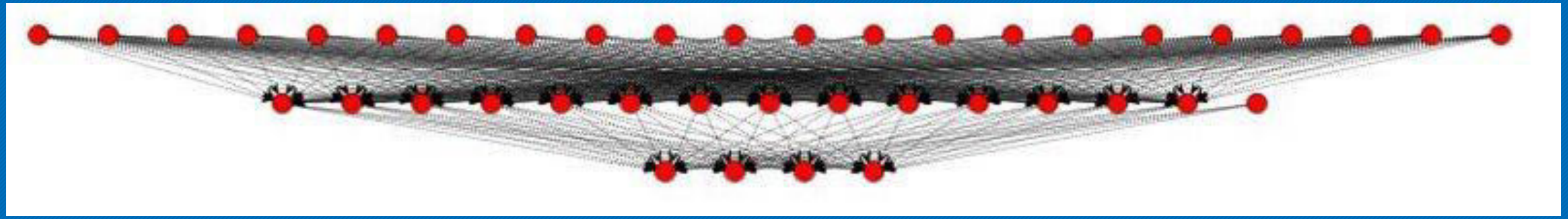
Transformed Dataset

buying: 1,0,0,0 instead of vhigh, 0,1,0,0 instead of high, 0,0,1,0 instead of med, 0,0,0,1 instead of low.
maint: 1,0,0,0 instead of vhigh, 0,1,0,0 instead of high, 0,0,1,0 instead of med, 0,0,0,1 instead of low.
doors: 0,0,0,1 instead of 2, 0,0,1,0 instead of 3, 0,1,0,0 instead of 4, 1,0,0,0 instead of 5more.
persons: 0,0,1 instead of 2, 0,1,0 instead of 4, 1,0,0 instead of more.
lug_boot: 0,0,1 instead of small, 0,1,0 instead of med, 1,0,0 instead of big.
safety: 0,0,1 instead of low, 0,1,0 instead of med, 1,0,0 instead of high.

Implementation

- Language used : Python
- Libraries used : Numpy, Keras
- Backend : Theano
- Type of Model : MLP(Multi-Layer Perceptron)

Network Architecture



Literature Survey

Classification Accuracy for Decision Trees

Percentage Split		Time in Seconds		Decision Tree	
Training %	Testing %	Build	Test	Correct %	Incorrect %
90	10	0.07	0.01	93.06	6.93
66	44	0.01	0.01	90.81	9.18
50	50	0.01	0.02	92.7	7.29
10 Folds		0.01	0.01	93.22	6.77

Literature Survey

Classification Accuracy for Naive Bayesian

Percentage Split		Time in Seconds		Naive Bayesian	
Training %	Testing %	Build	Test	Correct %	Incorrect %
0	10	0.02	0.05	93.06	6.93
66	44	0	0.03	92.51	7.48
50	50	0	0.04	92.7	7.29
10 Folds		0	0.25	93.51	6.48

Literature Survey

Classification Accuracy for Artificial Neural Network (ANN)

Percentage Split		Time in Seconds		ANN	
Training %	Testing %	Build	Test	Correct %	Incorrect %
90	10	7.1	0	93.06	6.93
66	44	7.19	0.01	90.81	9.18
50	50	6.98	0.02	92.7	7.29
10 Folds		7	0.03	93.51	6.48

Considerations:

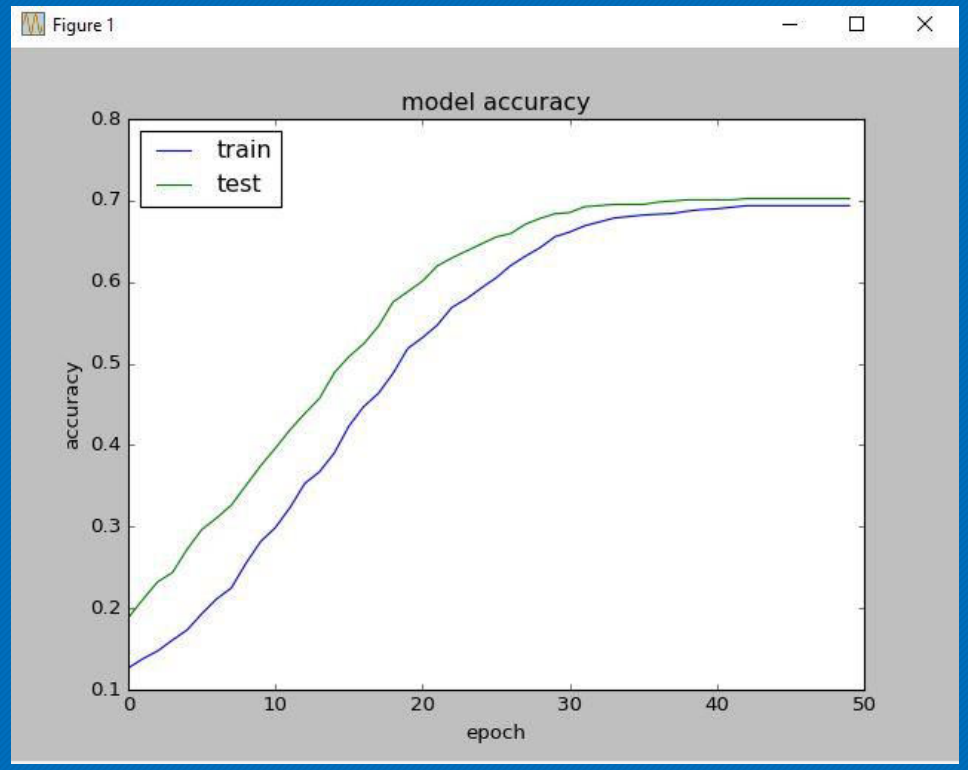
- CASE : 1
  - Input Neuron : 21
  - Number of Hidden layers:2
  - Hidden Neurons in first layer:100
  - Hidden Neurons in second layer:80
  - Output Neurons : 4
- Softmax function is used as activation function in both Hidden Layer as well as output layer.
- Learning Rate is considered to be : 0.01.
- Momentum :0.7
- Number of Epochs: 50

Results:

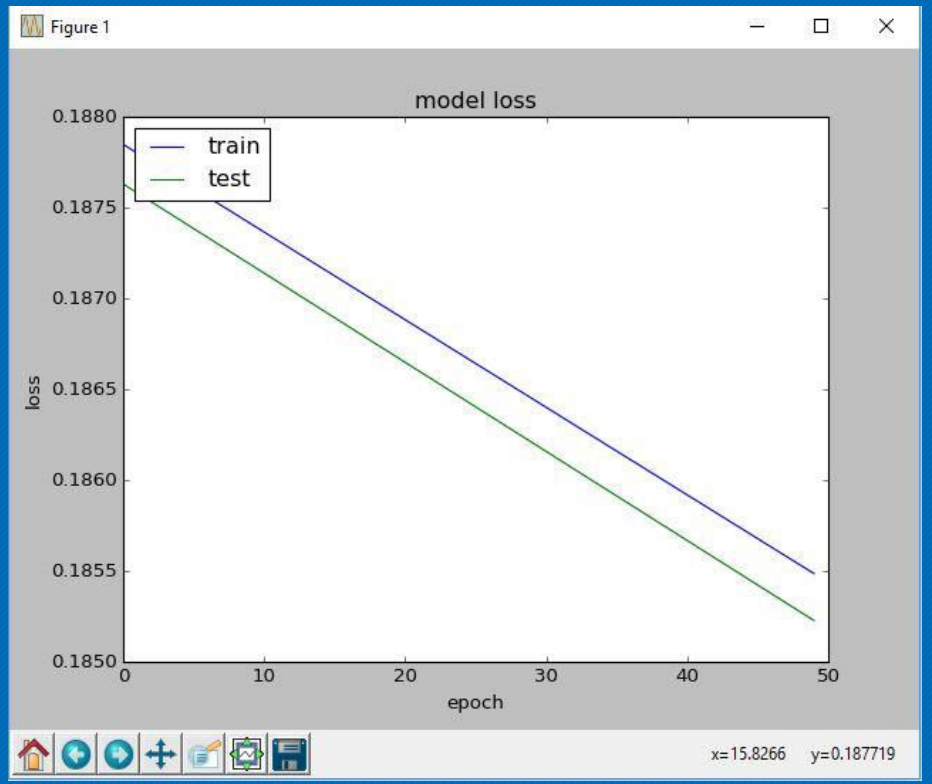
```
Epoch 48/50
1051/1051 [==============================] - 0s - loss: 0.1856 - acc: 0.6936 - val_loss: 0.1853 - val_acc: 0.7023
Epoch 49/50
1051/1051 [==============================] - 0s - loss: 0.1855 - acc: 0.6936 - val_loss: 0.1853 - val_acc: 0.7023
Epoch 50/50
1051/1051 [==============================] - 0s - loss: 0.1855 - acc: 0.6936 - val_loss: 0.1852 - val_acc: 0.7023
('mean squared error :', 0.18522654995959029)
('PREDICTED', array([[ 0.25511843,  0.25099114,  0.24640666,  0.24748382],
       [ 0.25787479,  0.25146687,  0.24438058,  0.24627775],
       [ 0.25605804,  0.25050306,  0.24628173,  0.24715714],
       ...,
       [ 0.25801155,  0.25079462,  0.24267635,  0.24851747],
       [ 0.25723404,  0.25106379,  0.2437406 ,  0.24796154],
       [ 0.25563523,  0.25113648,  0.24443632,  0.24879205]], dtype=float32))
('ORIGINAL', array([[1, 0, 0, 0],
       [1, 0, 0, 0],
       [1, 0, 0, 0],
       ...,
       [1, 0, 0, 0],
       [1, 0, 0, 0],
       [1, 0, 0, 0]]))
```

Fig(a) : Output with Accuracy and MSE

Graphical Representation :



Fig(b) : Accuracy Plot



Fig(c) : Error Plot

Considerations:

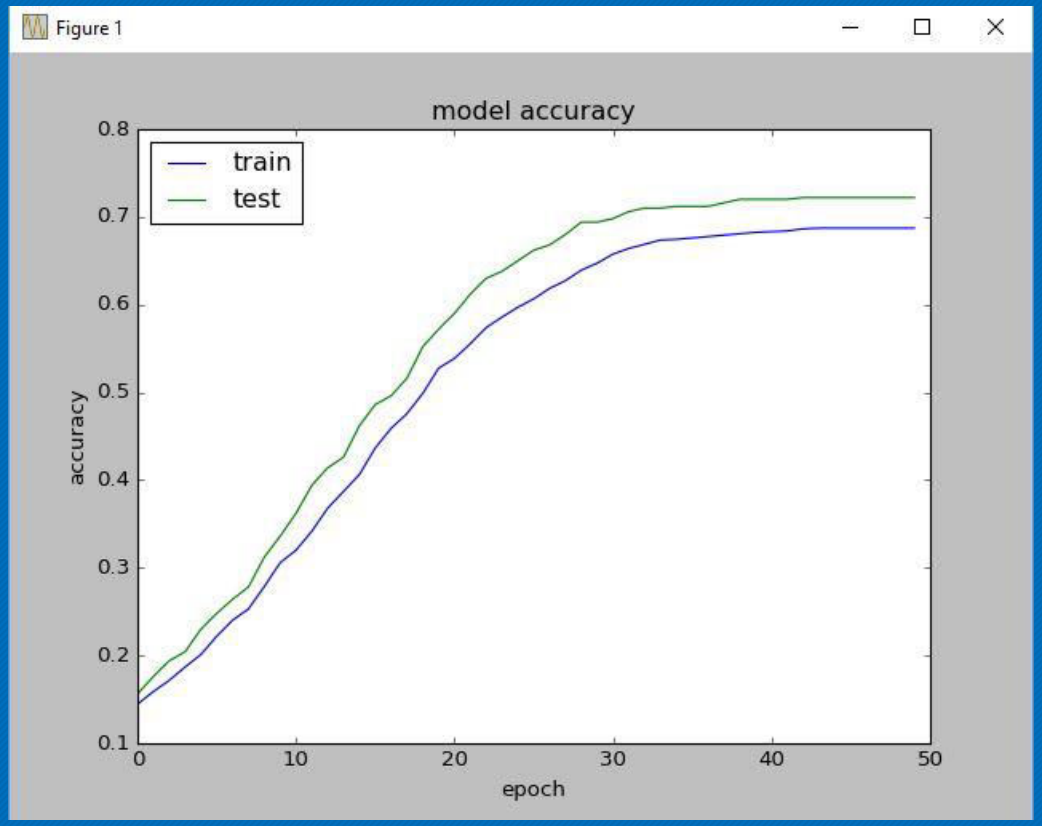
- CASE : 2
  - Input Neuron : 21
  - Number of Hidden layers:3
  - Neurons at hidden Layer 1: 100
  - Neurons at hidden Layer 2: 80
  - Neurons at hidden Layer 3:70
  - Output Neurons : 4
- Softmax function is used as activation function in both Hidden Layer as well as output layer.
- Learning Rate is considered to be : 0.01.
- Momentum :0.7
- Number of Epochs: 50

Results:

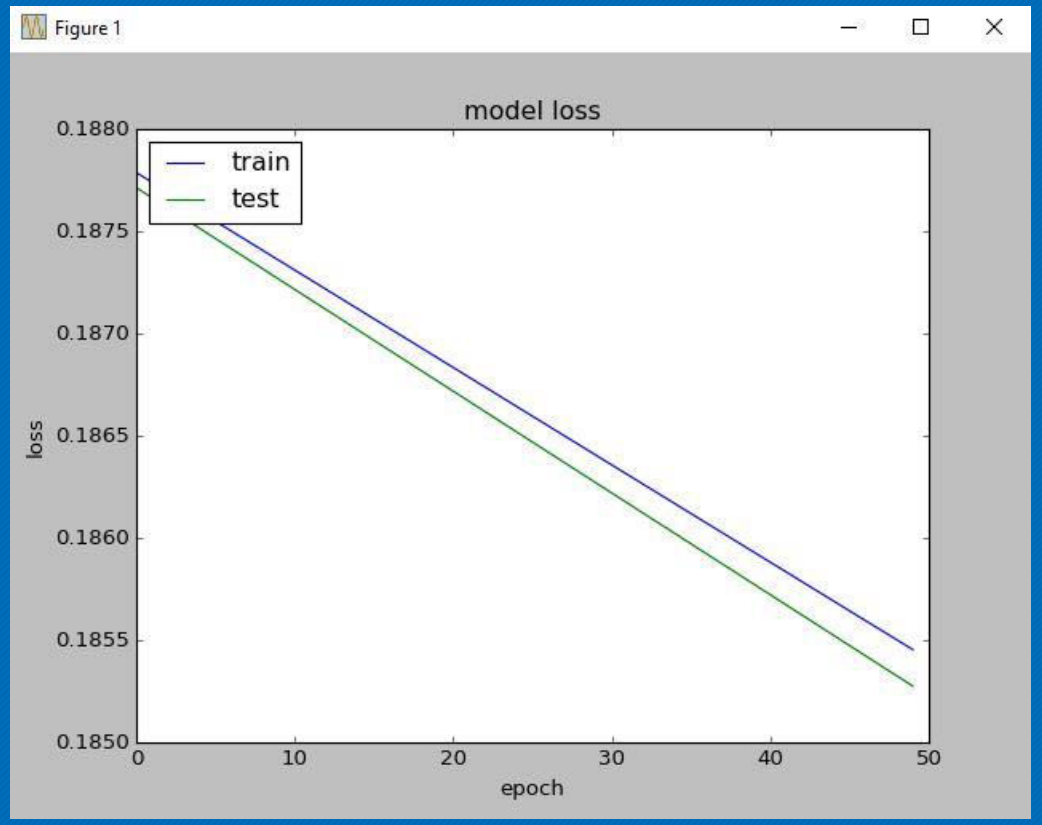
```
Epoch 48/50
1253/1253 [==============================] - 0s - loss: 0.1855 - acc: 0.6872 - val_loss: 0.1854 - val_acc: 0.7220
Epoch 49/50
1253/1253 [==============================] - 0s - loss: 0.1855 - acc: 0.6872 - val_loss: 0.1853 - val_acc: 0.7220
Epoch 50/50
1253/1253 [==============================] - 0s - loss: 0.1855 - acc: 0.6872 - val_loss: 0.1853 - val_acc: 0.7220
('mean squared error :', 0.18527450728416442)
('PREDICTED', array([[ 0.25498781,  0.25123912,  0.24630475,  0.24746837],
       [ 0.25774479,  0.25171679,  0.24427842,  0.24626009],
       [ 0.25592873,  0.25075242,  0.24617855,  0.24714026],
       ...,
       [ 0.25769141,  0.2488703 ,  0.24372566,  0.24971269],
       [ 0.2569291 ,  0.24913627,  0.24478082,  0.24915382],
       [ 0.2553277 ,  0.2492083 ,  0.24547216,  0.24999177]], dtype=float32))
('ORIGINAL', array([[1, 0, 0, 0],
       [1, 0, 0, 0],
       [1, 0, 0, 0],
       ...,
       [0, 0, 1, 0],
       [1, 0, 0, 0],
       [0, 0, 1, 0]]))
```

Fig(d) : Output with Accuracy and MSE

Graphical Representation :



Fig(e) : Accuracy Plot



Fig(f) : Error Plot

Considerations:

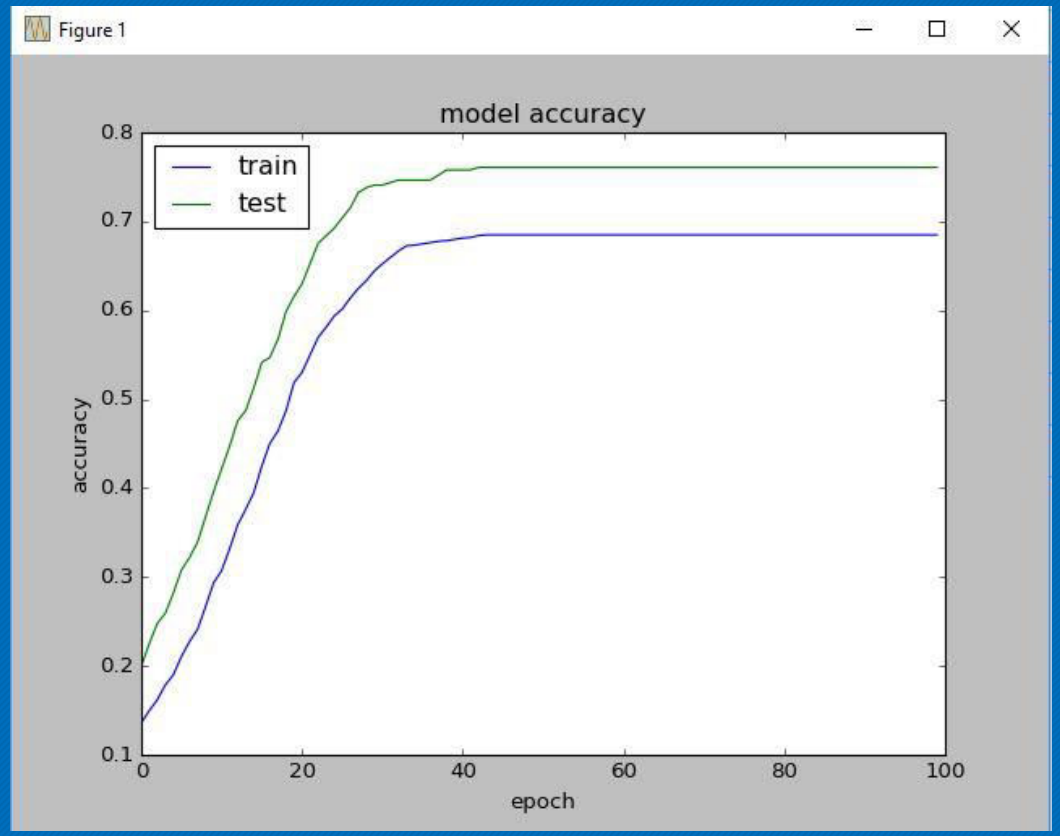
- CASE : 3
  - Input Neuron : 21
  - Number of Hidden layers:4
  - Neurons at hidden Layer 1:100
  - Neurons at hidden Layer 2: 80
  - Neurons at hidden Layer 3: 70
  - Neurons at hidden Layer 4:60
  - Output Neurons : 4
- Softmax function is used as activation function in both Hidden Layer as well as output layer.
- Learning Rate is considered to be : 0.01.
- Momentum :0.7
- Number of Epochs: 50

Results:

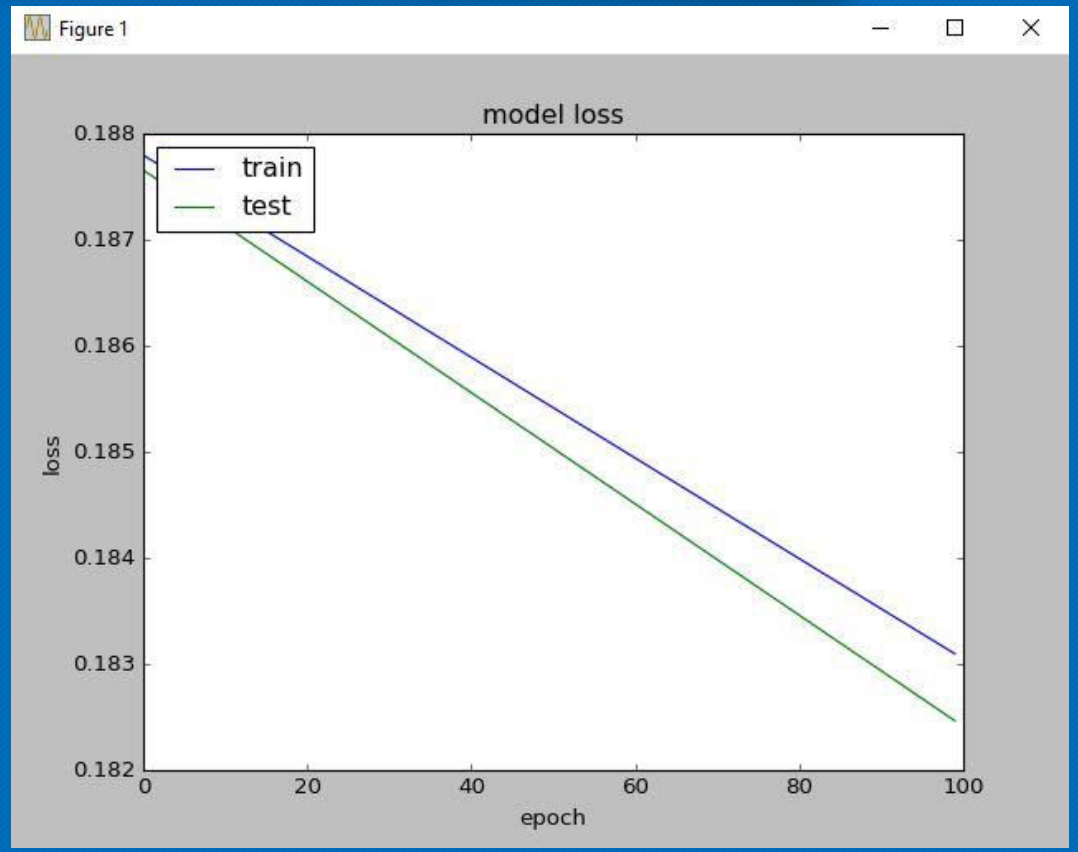
```
Epoch 98/100
1377/1377 [==============================] - 0s - loss: 0.1832 - acc: 0.6848 - val_loss: 0.1826 - val_acc: 0.7607
Epoch 99/100
1377/1377 [==============================] - 0s - loss: 0.1831 - acc: 0.6848 - val_loss: 0.1825 - val_acc: 0.7607
Epoch 100/100
1377/1377 [==============================] - 0s - loss: 0.1831 - acc: 0.6848 - val_loss: 0.1825 - val_acc: 0.7607
('mean squared error :', 0.18246412226277539)
('PREDICTED', array([[ 0.26247782,  0.25109023,  0.24260128,  0.2438307 ],
       [ 0.26528937,  0.25153652,  0.24056438,  0.24260977],
       [ 0.26358503,  0.25057003,  0.24240461,  0.24344037],
       ...,
       [ 0.26388296,  0.24856946,  0.24161132,  0.2459363 ],
       [ 0.26217005,  0.24867274,  0.24234924,  0.24680793],
       [ 0.26497576,  0.24912071,  0.24032374,  0.24557976]], dtype=float32))
('ORIGINAL', array([[1, 0, 0, 0],
       [1, 0, 0, 0],
       [1, 0, 0, 0],
       ...,
       [1, 0, 0, 0],
       [0, 0, 1, 0],
       [0, 0, 0, 1]]))
```

Fig(f) : Output with Accuracy and MSE

Graphical Representation :



Fig(h) : Accuracy Plot



Fig(i) : Error Plot

Considerations:

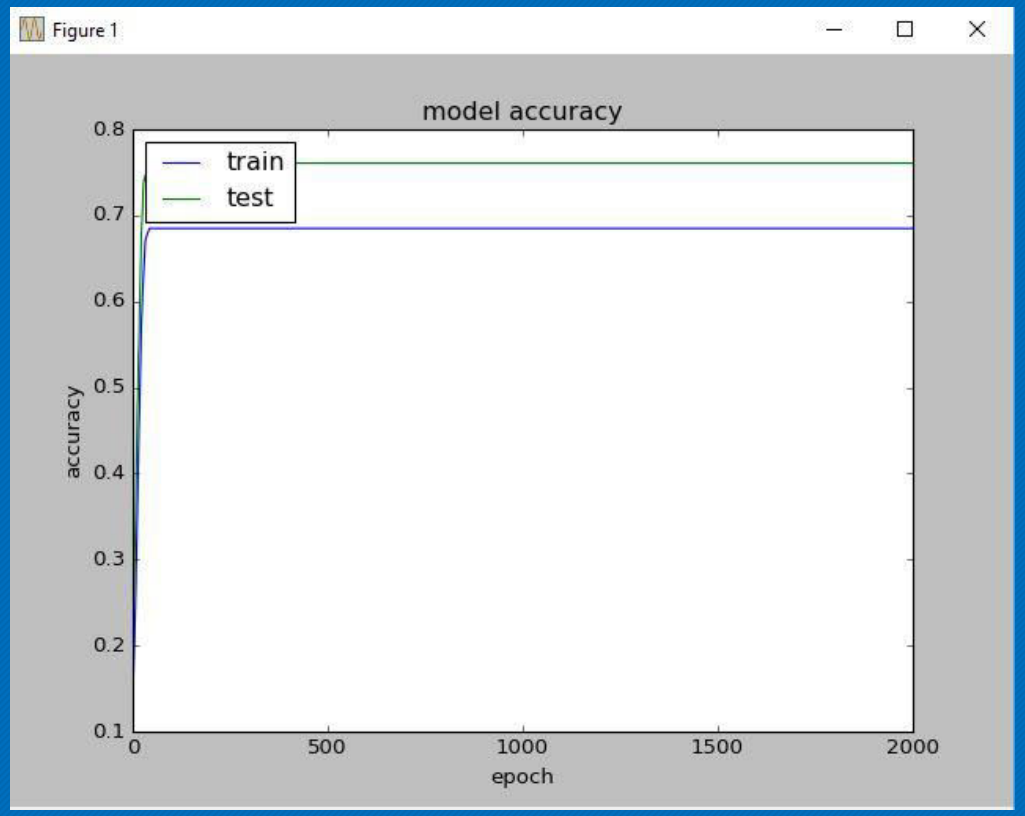
- CASE : 4
  - Input Neuron : 21
  - Number of Hidden layers:4
  - Neurons at hidden Layer 1:100
  - Neurons at hidden Layer 2: 80
  - Neurons at hidden Layer 3: 70
  - Neurons at hidden Layer 4 : 60
  - Output Neurons : 4
- Softmax function is used as activation function in both Hidden Layer as well as output layer.
- Learning Rate is considered to be : 0.01.
- Momentum :0.7
- Number of Epochs: 2000

Results:

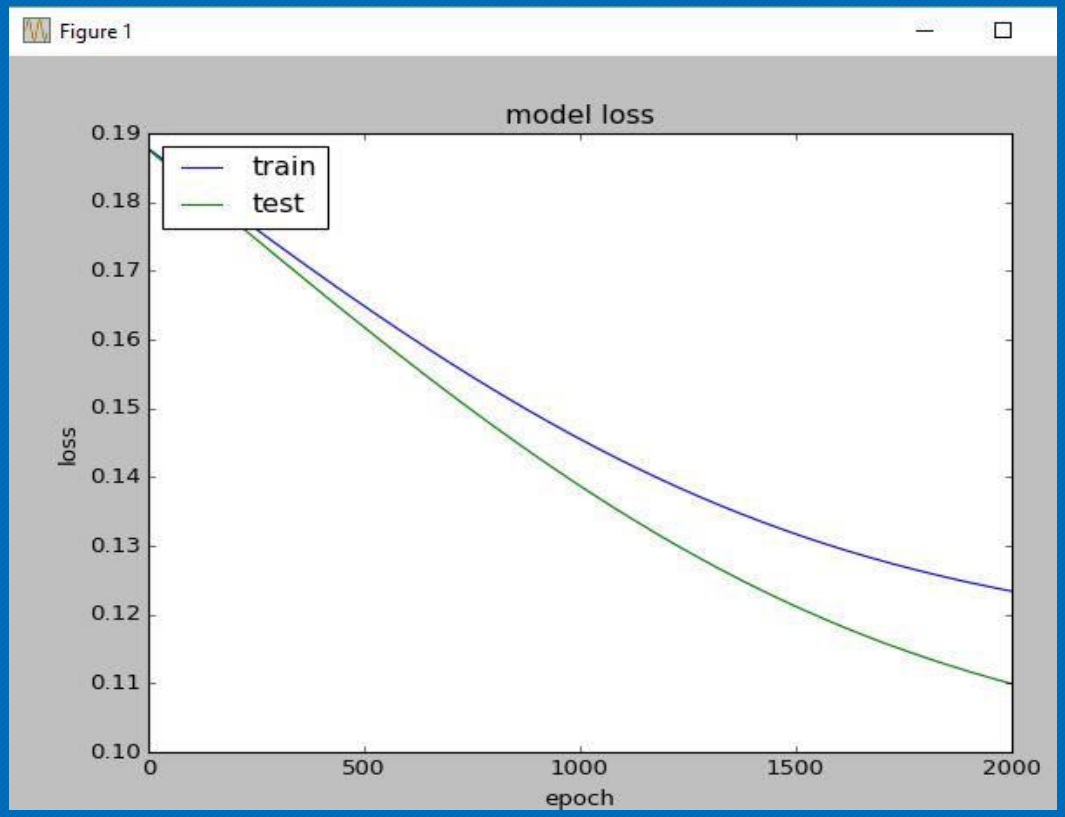
```
Epoch 1996/2000
1377/1377 [==============================] - 0s - loss: 0.1234 - acc: 0.6848 - val_loss: 0.1100 - val_acc: 0.7607
Epoch 1997/2000
1377/1377 [==============================] - 0s - loss: 0.1234 - acc: 0.6848 - val_loss: 0.1100 - val_acc: 0.7607
Epoch 1998/2000
1377/1377 [==============================] - 0s - loss: 0.1234 - acc: 0.6848 - val_loss: 0.1100 - val_acc: 0.7607
Epoch 1999/2000
1377/1377 [==============================] - 0s - loss: 0.1234 - acc: 0.6848 - val_loss: 0.1100 - val_acc: 0.7607
Epoch 2000/2000
1377/1377 [==============================] - 0s - loss: 0.1234 - acc: 0.6848 - val_loss: 0.1099 - val_acc: 0.7607
('mean squared error :', 0.10994575832813894)
('PREDICTED', array([[ 0.57206655,  0.18295768,  0.1211511 ,  0.12382458],
       [ 0.57741946,  0.1815847 ,  0.1189913 ,  0.12200451],
       [ 0.59014457,  0.17648116,  0.11531767,  0.11805657],
       ...,
       [ 0.56895602,  0.18234132,  0.12239027,  0.12631236],
       [ 0.54921758,  0.18871984,  0.12910526,  0.13295737],
       [ 0.55484444,  0.1873481 ,  0.12680177,  0.13100566]], dtype=float32))
('ORIGINAL', array([[1, 0, 0, 0],
       [1, 0, 0, 0],
       [1, 0, 0, 0],
       ...,
       [1, 0, 0, 0],
       [0, 0, 1, 0],
       [0, 0, 0, 1]]))
```

Fig(f) : Output with Accuracy and MSE

Graphical Representation :



Fig(h) : Accuracy Plot



Fig(i) : Error Plot

Future Work:

- We have to add more data items to get more accurate results.
- Need more Optimization Techniques.
- Can use more complex networks.

References:

- Knowledge acquisition and explanation for multi-attribute decision making by M Bohanec, V Rajkovic.
- Machine Learning by Function Decomposition Blaz Zupan, Marko Bohanec, Ivan Bratko, Janez Demsar.
- https://archive.ics.uci.edu/ml/datasets/Car+Evaluation.
- http://neuroph.sourceforge.net/tutorials/carevaluation1/carevaluation1.html