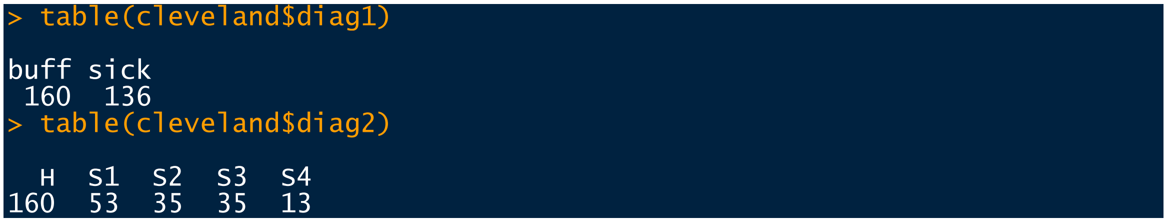
Classification of Heart Disease

Classification – Using Neural Network, CART, Random Forest

Dataset: Cleveland Heart Disease Study
296 observations and 15 variables

Response Variable: diag1
Buff- not sick with heart disease
Sick-sick with heart disease

We will be removing the other response variable from the dataset – diag2 as it shows the stage of heart disease which will have influence on prediction as its corelated to response variable directly.

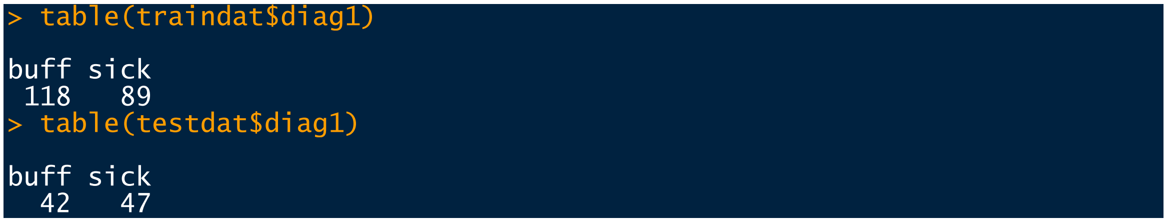


```
> table(cleveland$diag1)

buff sick
 160  136
> table(cleveland$diag2)

  H  S1  S2  S3  S4
160  53  35  35  13
```

Dividing the dataset into test and train data.



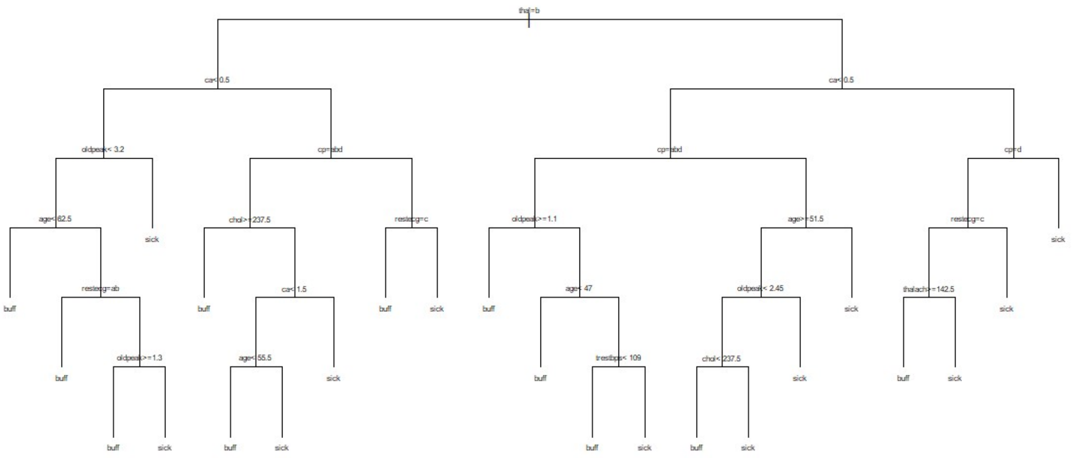
```
> table(traindat$diag1)

buff sick
 118   89
> table(testdat$diag1)

buff sick
  42   47
```

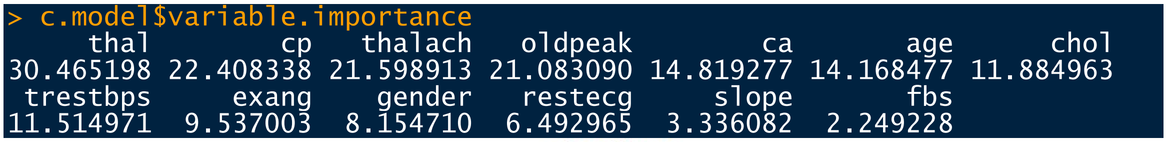
CART:

Classification tree for the given dataset which will take the predictor which has better prediction rate of response variable and keeps splitting and growing the tree to the max depth of 30. CART models are easily interpretable as each split will have predictor and value used for split.

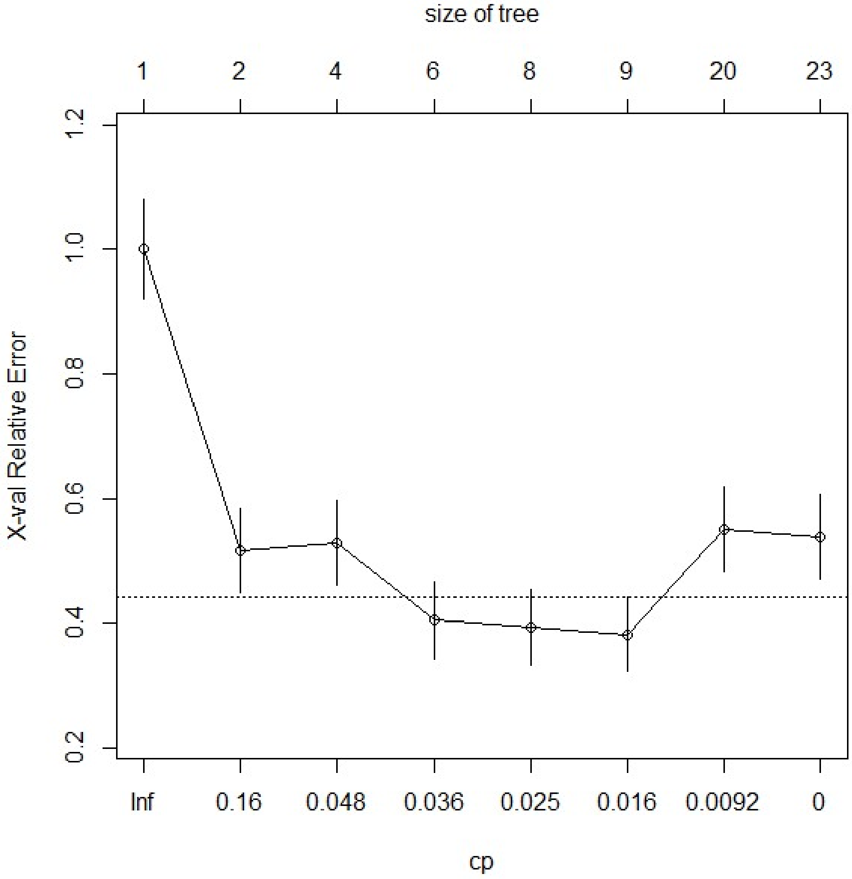


Full tree always overfits data and has poor performance on the test data so we have to prune tree to optimal depth so we can have better performance on the test data.

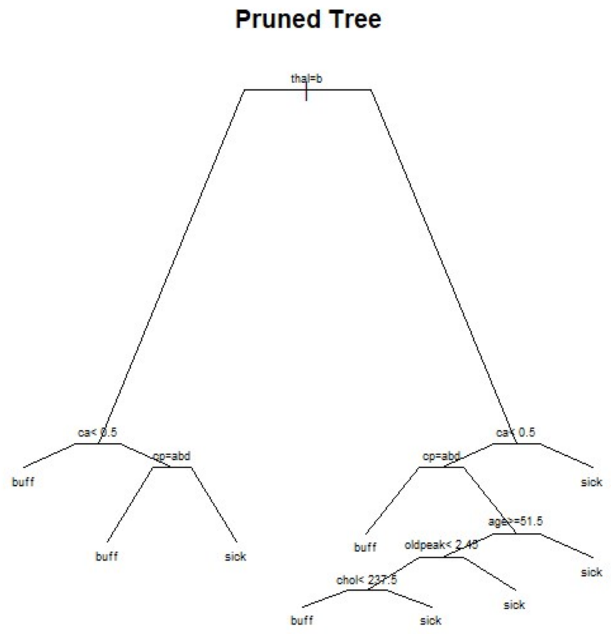
Variable Importance:

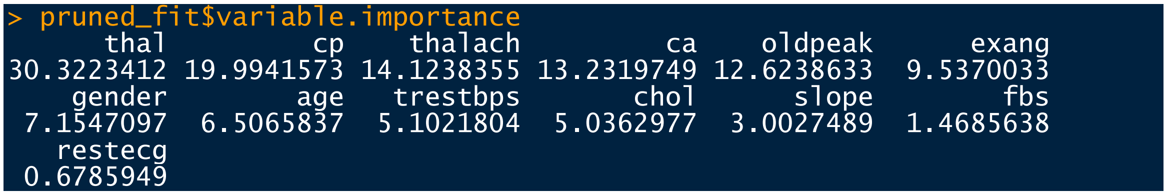


```
> c.model$variable.importance
     thal        cp   thalach   oldpeak        ca       age      chol
30.465198 22.408338 21.598913 21.083090 14.819277 14.168477 11.884963
 trestbps     exang    gender   restecg     slope       fbs
11.514971  9.537003  8.154710  6.492965  3.336082  2.249228
```



We have complexity parameter that shows how model improves over each split. So, we see when tree size is 9 we have relative error really less so this will be optimal depth of the classification tree and the CP value is the optimal value. So, we will prune the tree to this depth.





```
> pruned_fit$variable.importance
      thal         cp    thalach         ca    oldpeak      exang 
30.3223412 19.9941573 14.1238355 13.2319749 12.6238633  9.5370033 
    gender        age   trestbps       chol      slope        fbs 
 7.1547097  6.5065837  5.1021804  5.0362977  3.0027489  1.4685638 
   restecg 
 0.6785949 
```

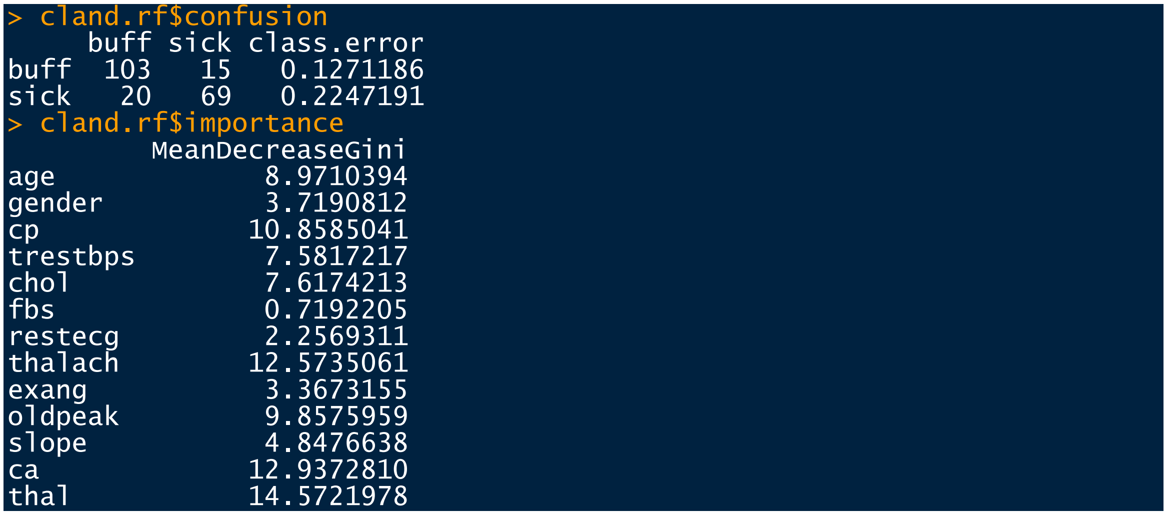
Now let’s verify how test and train data performances on both the models:

Model Type	Train Accuracy	Test Accuracy
Full Tree	95.56	74.16
Pruned tree-optimal	**89.37**	**83.15**

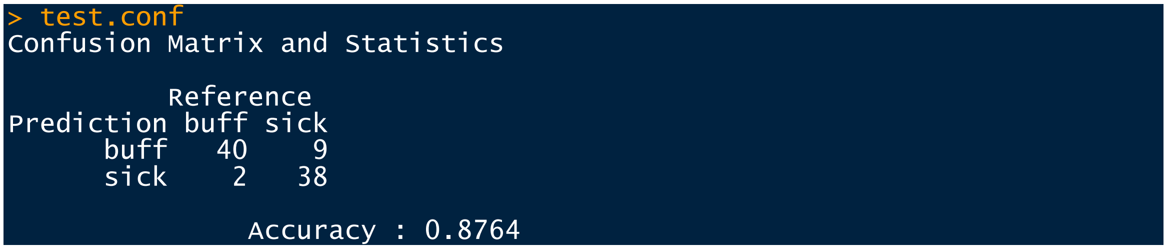
We can observe with full tree there is clear sign of overfitting as model performs well on train data but not on test data.

Random Forest:

Create a random sample from the training data and for each sample build a tree with specific number of features which are chosen randomly resulting in the multiple unique trees so it’s called random forest. For regression, the results of all trees are averaged and for the classification based on number of votes of results of trees the result is derived.



```
> cland.rf$confusion
     buff sick class.error
buff  103   15   0.1271186
sick   20   69   0.2247191
> cland.rf$importance
         MeanDecreaseGini
age             8.9710394
gender          3.7190812
cp             10.8585041
trestbps        7.5817217
chol            7.6174213
fbs             0.7192205
restecg         2.2569311
thalach        12.5735061
exang           3.3673155
oldpeak         9.8575959
slope           4.8476638
ca             12.9372810
thal           14.5721978
```



```
> test.conf
Confusion Matrix and Statistics

          Reference
Prediction buff sick
      buff   40    9
      sick    2   38

               Accuracy : 0.8764
```

As random forest consists of multiple trees we can't interpret model but variable importance does help to understand which predictor does have influence on the response variable. But accuracy is better than **CART** as it is the result of multiple trees and generalization is more.

Neural Networks:

Neural Network is a supervised learning system built of a large number of simple elements called neurons and perceptrons. Each neuron learns from input and feeds it results to next layer. Simple neural network consists of single layer of hidden layers and deep neural networks consists of multiple neural networks which computationally intensive. Neural Networks are efficient if given larger dataset and computing power.

<u>Forward Pass:</u>

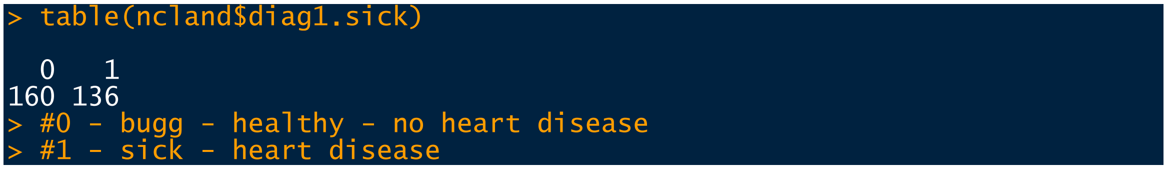
Input data is passed thru the first layer with initialized weights and then again from hidden layer to the output layer other set of weights and using the activation function it results an output

<u>Back Propagation:</u>

The error is calculated and then derivative of error wrt to each layer of weights is updated using the learning rate and if learning rate is large we might miss the global minimum and is learning rate is small it learns slow.

Both forward pass and back propagation continues till all weights are learnt. Then those weights are used for prediction of test data. Neural Networks are complex due to so many parameters to tune and its computing intensive.

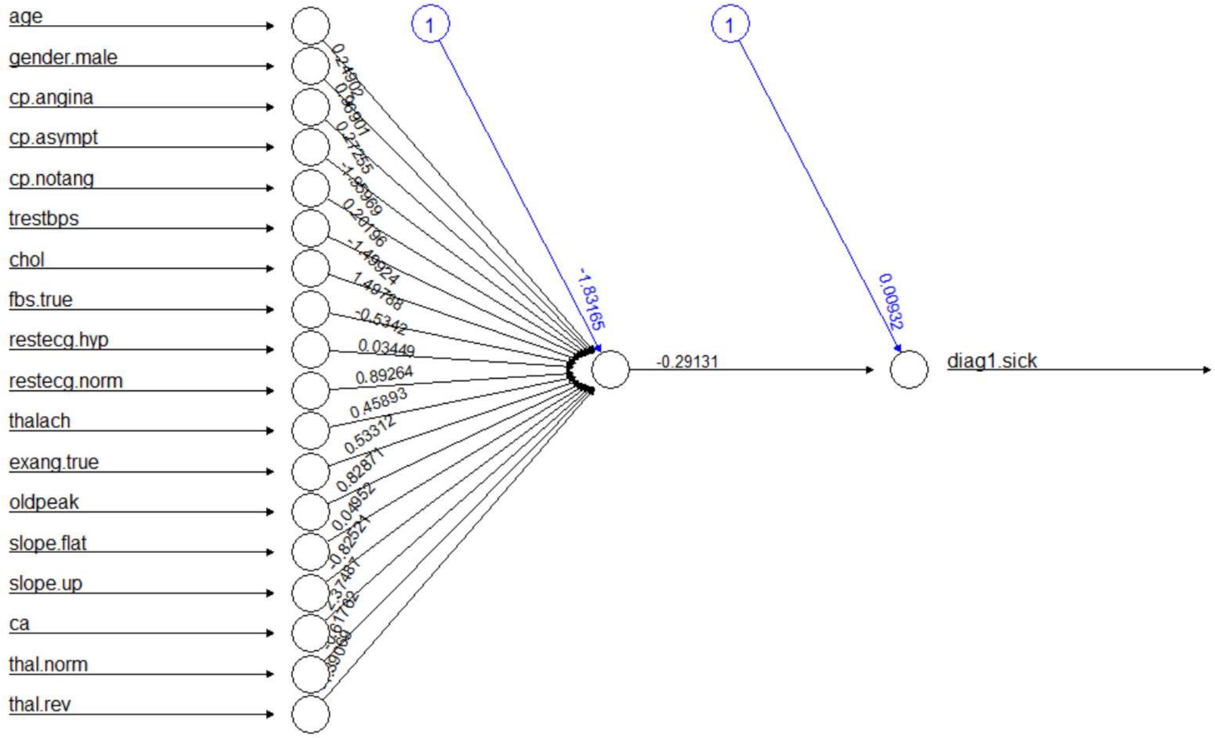
Note: Neural Network accepts just numerical data so any categorical values to be converted to factor of numeric values. So, for Cleveland data we did convert all categorical data to numeric data.



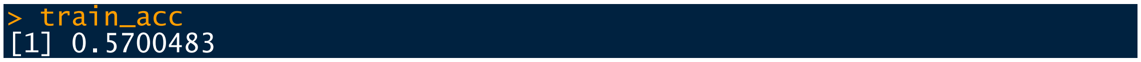
```
> table(ncland$diag1.sick)

  0   1 
160 136 
> #0 - bugg - healthy - no heart disease
> #1 - sick - heart disease
```

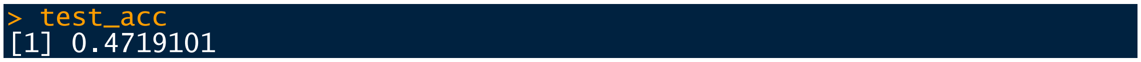
Neural Net – One hidden layer:



Let's verify its prediction accuracy just with one hidden layer.

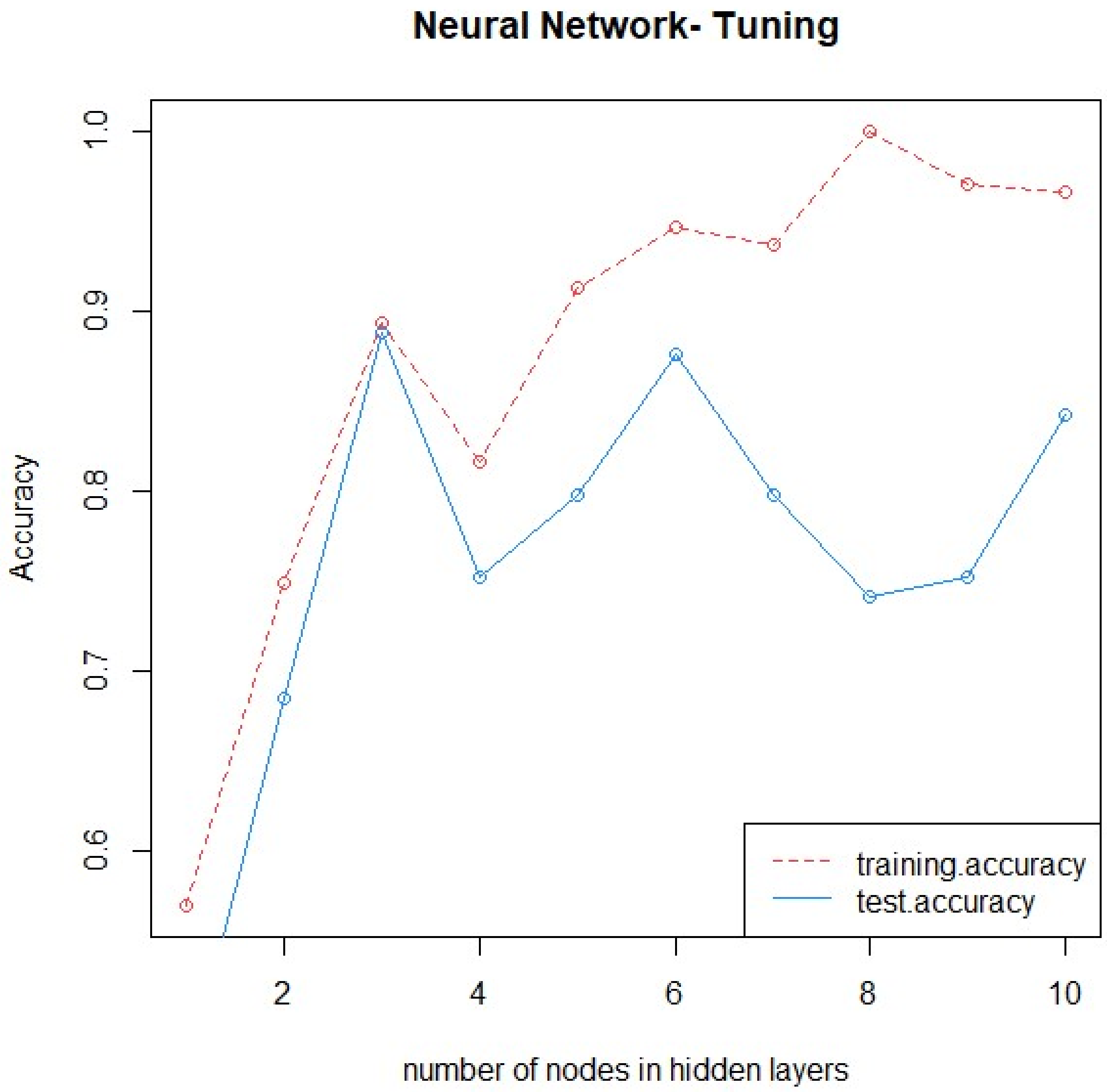


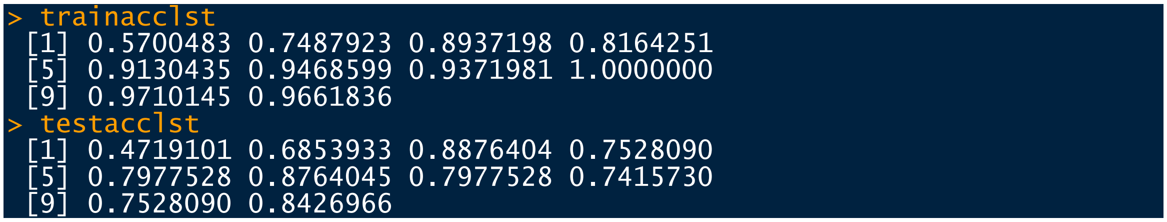
```
> train_acc
[1] 0.5700483
```



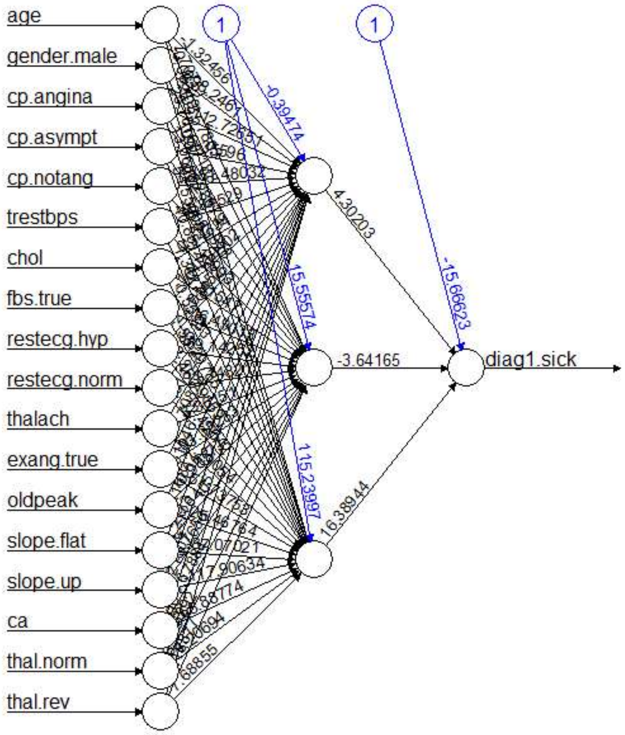
```
> test_acc
[1] 0.4719101
```

Accuracy for both train and test data is bad so we have to tune number of hidden layers to learn weights to the mark and get better results of neural networks.

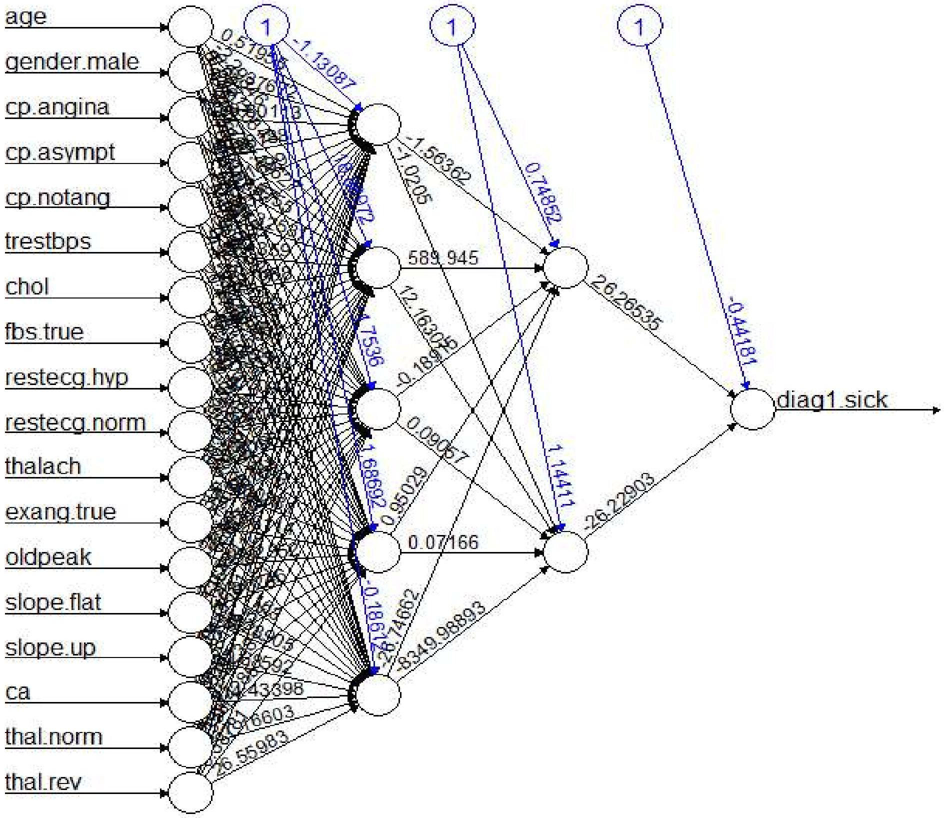



```
> trainacclst
 [1] 0.5700483 0.7487923 0.8937198 0.8164251
 [5] 0.9130435 0.9468599 0.9371981 1.0000000
 [9] 0.9710145 0.9661836
> testacclst
 [1] 0.4719101 0.6853933 0.8876404 0.7528090
 [5] 0.7977528 0.8764045 0.7977528 0.7415730
 [9] 0.7528090 0.8426966
```

Optimal number of nodes in single hidden layer is 3 for which the test and train accuracy is more.



Deep Neural Net: Increasing number of hidden layers



Conclusion:

Model Type	Test Data Performance	Parameters to learn
CART	83.15	Depth of tree
Random Forest	87.64	Number of trees, maximum number of features for each split, maximum depth
Neural Network Tuned	88.76	weights, learning rate, number of nodes
Deep Net-not tuned	71.91	weights, learning rate, number of nodes, number of hidden layers

Neural network is efficient given input dataset is large as it’s a slow learner and computationally intensive. But its complex to interpret as its black box model due to number of layers and number of parameters it makes model difficult to understand and interpret compared to CART and Random Forest.