**Assignment-Neural Networks**

1. Concrete Data set:

Problem Statement- Prepare a model for strength of concrete data using Neural Networks

Soln: #apply normalization to entire data frame

concrete

concrete\_norm<-as.data.frame(lapply(concrete,normalize))

#Create Training and testing data

Concrete\_train<-concrete\_norm[1:773,]

Concrete\_test<-concrete\_norm[774:1030,]

#train the neural net model

library(neuralnet)

concrete\_model<-neuralnet(formula=strength~cement+slag+ash+water

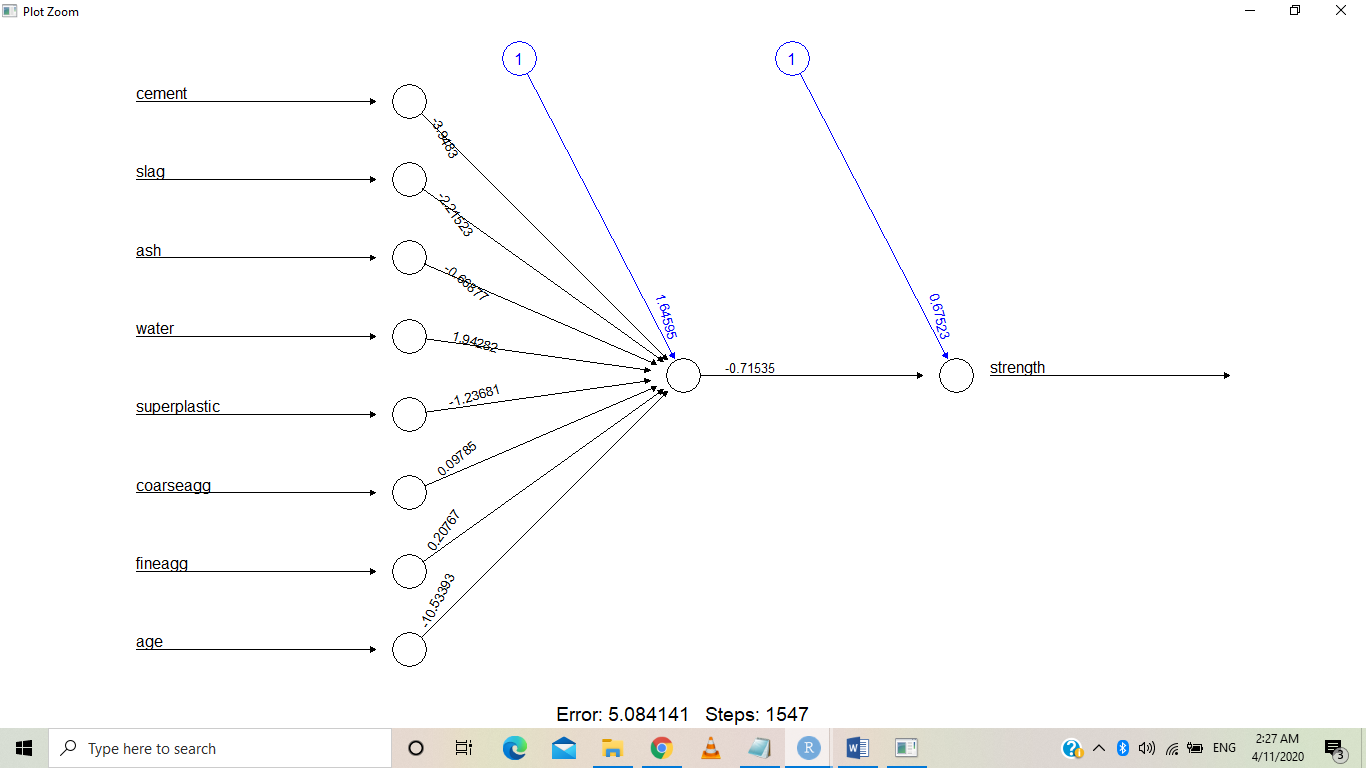
+superplastic+coarseagg+fineagg+

age,data=Concrete\_train)

#visualize the network topology

windows()

plot(concrete\_model)



Steps: 1547

Error: 5.084

#Evaluate model Performance

model\_results<-compute(concrete\_model,Concrete\_test[1:8])

#Obtain predicted strength values

predicted\_strength<-model\_results$net.result

#examine corelation between predicted and actual values

cor(predicted\_strength, Concrete\_test$strength)

0.8057751

80% corelateion found between predicted value and calculated value

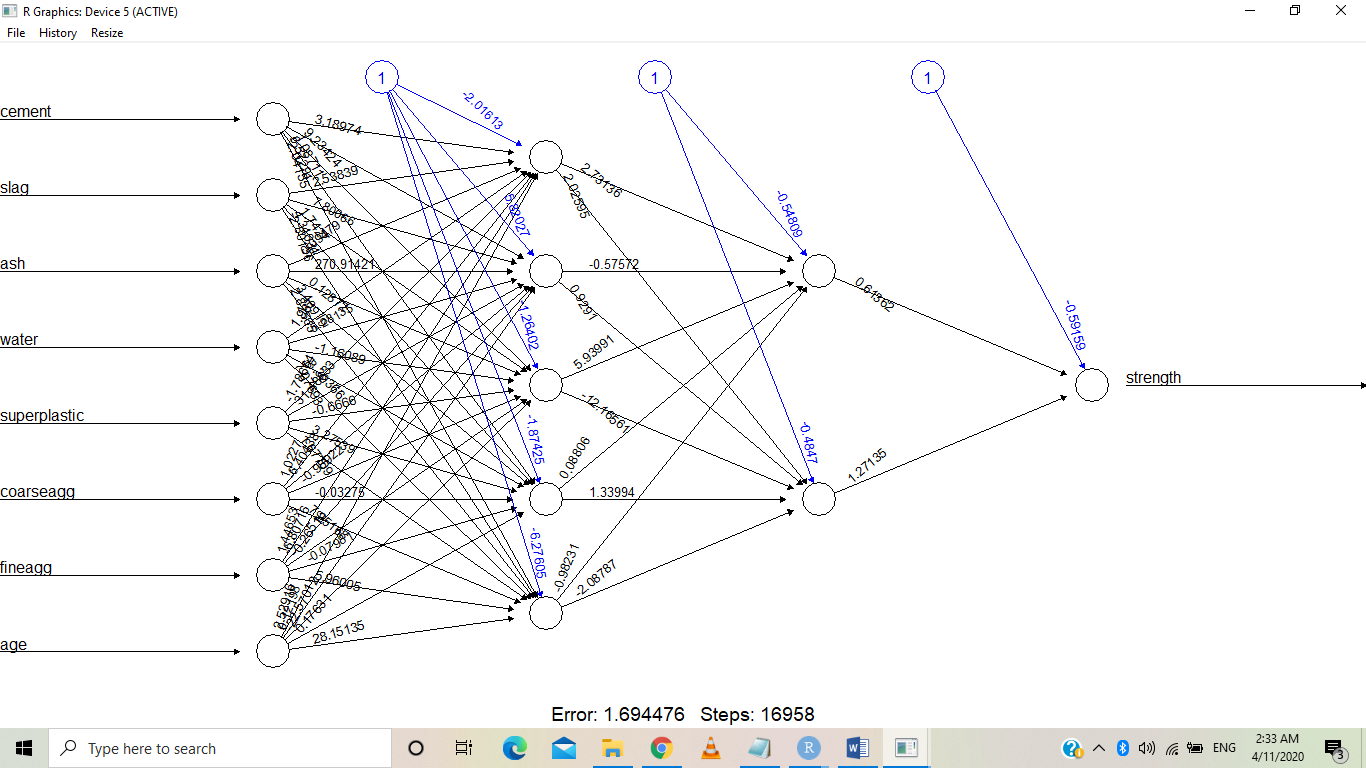
Reiterating the model with hidden neurons for further improving model performance;

concrete\_model2<-neuralnet(formula=strength~cement+slag+ash+water

+superplastic+coarseagg+fineagg+

age,data=Concrete\_train,hidden=c(5,2))

plot(concrete\_model2)



#Evaluate results as did before

model\_results2<-compute(concrete\_model2,concrete\_test[1:8])

predicted\_strength2<-model\_results2$net.result

cor(predicted\_strength2, concrete\_test$strength)

0.9135928

Performance improved and errors reduced.

1. 50 Startups:

Problem statement: Build a Neural Network model for 50\_startups data to predict profit

#50 Startups

#load the data

startup <- `50\_Startups`

#creating dummy variable for "State" column

startup <-dummy\_cols(startup,select\_columns =='State', remove\_selected\_column =T)

View(startup)

#custom normalize the data

normalize <- function(x) {

return ((x - min(x))/(max(x)-min(x)))

}

startup\_nl <- as.data.frame(lapply(startup[1:4], normalize))

View(startup\_nl)

startup\_norm <- cbind(startup\_nl, startup[,c(5,6,7)])

View(startup\_norm)

#creating training and testing datasets

start\_train <- startup\_norm[1:40,]

start\_test <- startup\_norm[41:50,]

#training model without hidden layers

startup\_model <- neuralnet(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend , data = start\_train)

windows()

plot(startup\_model)

#obtaining model result

startup\_model\_result <- compute(startup\_model, start\_test[c(1,2,3,5,6,7)])

#predicted strength vs actual strength

start\_pred <- startup\_model\_result$net.result

#correlation between predicted vs actual

cor(start\_pred, start\_test$Profit) #75.39

#training model with hidden layers

startup\_model2 <- neuralnet(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend , data = start\_train, hidden = c(5,3))

windows()

plot(startup\_model2)

#model result

startup\_model\_result2 <- compute(startup\_model2, start\_test[,c(1,2,3,5,6,7)])

#predicted vs actual

start\_pred2 <- startup\_model\_result2$net.result

#correlation between pedicted and actual

cor(start\_pred2, start\_test$Profit) #75.8

#experiments with hidden layers and feature selection

sm <- neuralnet(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend , data = start\_train, hidden = c(10,5))

windows()

plot(sm)

sm\_result <- compute(sm, start\_test[c(1,2,3,5,6,7)])

sm\_pred <- sm\_result$net.result

cor(sm\_pred, start\_test$Profit) #73.01

#increase in hidden layers has led to decrease in accuracy

#building model with a lesser hidden layer

sm2 <- neuralnet(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend , data = start\_train, hidden = c(3,2))

plot(sm2)

sm2\_result <- compute(sm2, start\_test[-4])

sm2\_pred <- sm2\_result$net.result

cor(sm2\_pred, start\_test$Profit) #74.9

sm3 <- neuralnet(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend , data = start\_train, hidden = c(4,1))

sm3\_result <- compute(sm3, start\_test[-4])

sm3\_pred <- sm3\_result$net.result

cor(sm3\_pred, start\_test$Profit) #78.79

sm4 <- neuralnet(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend , data = start\_train, hidden = c(6,1))

sm4\_result <- compute(sm4, start\_test[-4])

sm4\_pred <- sm4\_result$net.result

cor(sm4\_pred, start\_test$Profit) #79.72

1. PREDICT THE BURNED AREA OF FOREST FIRES WITH NEURAL NETWORKS

#loading the data

forestfires<-forestfires

forestfires$size\_category <- as.numeric(forestfires$size\_category)

ff <- forestfires[,-c(1,2)]

View(ff)

is.null(ff) #checking if the dataset has null values

#normalize the dataset

normalize <- function(x) {

return ((x - min(x))/ (max(x)-min(x)))}

ff\_norm <- as.data.frame(lapply(ff, normalize))

#EDA

#scatter plot matrix

plot(ff$area, type = "l")

windows()

pairs(ff\_norm)

cor(ff)

ff\_eda <- ff\_norm[,1:9]

##checking for multi-collinearity

install.packages("car")

library(car)

ff\_mlrmod <- lm(ff\_eda$area~. , data = ff\_eda)

car::vif(ff\_mlrmod)

pairs(ff\_eda)

cor(ff\_eda)

#subset selection

library(MASS)

stepAIC(ff\_mlrmod)# the StepAIC value is the least at 2954.8 except MonthSep and Daywed

#creating training and testing datasets

library(C50)

library(caret)

intraining <- createDataPartition(ff\_norm$area, p = 0.8, list = F)

ff\_train <- ff\_norm[intraining,]

ff\_test <- ff\_norm[-intraining,]

#building model without hidden layers

library(neuralnet)

ff\_mod <- neuralnet(formula = area ~ FFMC + DMC + DC + ISI + temp + RH + wind + rain

, data = ff\_train)

ff\_mod <- neuralnet(formula = area ~ FFMC + DMC + DC + ISI + temp + RH + wind + rain +

dayfri + daymon + daysat + daysun + daythu + daytue + monthapr +

monthaug + monthdec + monthfeb + monthjan + monthjul + monthjun +

monthmar + monthmay + monthnov + monthoct + size\_category, data = ff\_train )

ff\_mod\_result <- compute(ff\_mod, ff\_test[-9])

ff\_pred <- ff\_mod\_result$net.result

cor(ff\_pred, ff\_test$area)

Result: [1,] 0.4104754

