**Prepare a model for strength of concrete data using Neural Networks**

**Ans:**

> library(neuralnet)

> library(NeuralNetTools)

> library(nnet)

> library(caret)

> library(corrplot)

> library(GGally)

> concrete <- read.csv(file.choose())

> head(concrete)

cement slag ash water superplastic coarseagg fineagg age strength

1 141.3 212.0 0.0 203.5 0.0 971.8 748.5 28 29.89

2 168.9 42.2 124.3 158.3 10.8 1080.8 796.2 14 23.51

3 250.0 0.0 95.7 187.4 5.5 956.9 861.2 28 29.22

4 266.0 114.0 0.0 228.0 0.0 932.0 670.0 28 45.85

5 154.8 183.4 0.0 193.3 9.1 1047.4 696.7 28 18.29

6 255.0 0.0 0.0 192.0 0.0 889.8 945.0 90 21.86

> str(concrete)

'data.frame': 1030 obs. of 9 variables:

$ cement : num 141 169 250 266 155 ...

$ slag : num 212 42.2 0 114 183.4 ...

$ ash : num 0 124.3 95.7 0 0 ...

$ water : num 204 158 187 228 193 ...

$ superplastic: num 0 10.8 5.5 0 9.1 0 0 6.4 0 9 ...

$ coarseagg : num 972 1081 957 932 1047 ...

$ fineagg : num 748 796 861 670 697 ...

$ age : int 28 14 28 28 28 90 7 56 28 28 ...

$ strength : num 29.9 23.5 29.2 45.9 18.3 ...

> summary(concrete)

cement slag ash water superplastic coarseagg

Min. :102.0 Min. : 0.0 Min. : 0.00 Min. :121.8 Min. : 0.000 Min. : 801.0

1st Qu.:192.4 1st Qu.: 0.0 1st Qu.: 0.00 1st Qu.:164.9 1st Qu.: 0.000 1st Qu.: 932.0

Median :272.9 Median : 22.0 Median : 0.00 Median :185.0 Median : 6.400 Median : 968.0

Mean :281.2 Mean : 73.9 Mean : 54.19 Mean :181.6 Mean : 6.205 Mean : 972.9

3rd Qu.:350.0 3rd Qu.:142.9 3rd Qu.:118.30 3rd Qu.:192.0 3rd Qu.:10.200 3rd Qu.:1029.4

Max. :540.0 Max. :359.4 Max. :200.10 Max. :247.0 Max. :32.200 Max. :1145.0

fineagg age strength

Min. :594.0 Min. : 1.00 Min. : 2.33

1st Qu.:731.0 1st Qu.: 7.00 1st Qu.:23.71

Median :779.5 Median : 28.00 Median :34.45

Mean :773.6 Mean : 45.66 Mean :35.82

3rd Qu.:824.0 3rd Qu.: 56.00 3rd Qu.:46.13

Max. :992.6 Max. :365.00 Max. :82.60

> attach(concrete)

**Normalising function**

> normm <- function(x){

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

}

**Normalising df**

> norm\_concrete <- as.data.frame(lapply(concrete,normm))

> head(norm\_concrete)

cement slag ash water superplastic coarseagg fineagg

1 0.08972603 0.5898720 0.0000000 0.6525559 0.0000000 0.4965116 0.3876066

2 0.15273973 0.1174179 0.6211894 0.2915335 0.3354037 0.8133721 0.5072755

3 0.33789954 0.0000000 0.4782609 0.5239617 0.1708075 0.4531977 0.6703462

4 0.37442922 0.3171953 0.0000000 0.8482428 0.0000000 0.3808140 0.1906673

5 0.12054795 0.5102949 0.0000000 0.5710863 0.2826087 0.7162791 0.2576518

6 0.34931507 0.0000000 0.0000000 0.5607029 0.0000000 0.2581395 0.8805820

age strength

1 0.07417582 0.3433412

2 0.03571429 0.2638595

3 0.07417582 0.3349944

4 0.07417582 0.5421702

5 0.07417582 0.1988290

6 0.24450549 0.2433038

**Splitting of data to test and train**

> trainn <- createDataPartition(norm\_concrete$strength,p=0.8,list = F)

> trn\_c <- norm\_concrete[trainn,]

> tst\_c <- norm\_concrete[-trainn,]

**Model building**

> model\_con <- neuralnet(strength~.,data = trn\_c)

> str(model\_con)

List of 14

$ call : language neuralnet(formula = strength ~ ., data = trn\_c)

$ response : num [1:826, 1] 0.343 0.264 0.335 0.542 0.199 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:826] "1" "2" "3" "4" ...

.. ..$ : chr "strength"

$ covariate : num [1:826, 1:8] 0.0897 0.1527 0.3379 0.3744 0.1205 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:826] "1" "2" "3" "4" ...

.. ..$ : chr [1:8] "cement" "slag" "ash" "water" ...

$ model.list :List of 2

..$ response : chr "strength"

..$ variables: chr [1:8] "cement" "slag" "ash" "water" ...

$ err.fct :function (x, y)

..- attr(\*, "type")= chr "sse"

$ act.fct :function (x)

..- attr(\*, "type")= chr "logistic"

$ linear.output : logi TRUE

$ data :'data.frame': 826 obs. of 9 variables:

..$ cement : num [1:826] 0.0897 0.1527 0.3379 0.3744 0.1205 ...

..$ slag : num [1:826] 0.59 0.117 0 0.317 0.51 ...

..$ ash : num [1:826] 0 0.621 0.478 0 0 ...

..$ water : num [1:826] 0.653 0.292 0.524 0.848 0.571 ...

..$ superplastic: num [1:826] 0 0.335 0.171 0 0.283 ...

..$ coarseagg : num [1:826] 0.497 0.813 0.453 0.381 0.716 ...

..$ fineagg : num [1:826] 0.388 0.507 0.67 0.191 0.258 ...

..$ age : num [1:826] 0.0742 0.0357 0.0742 0.0742 0.0742 ...

..$ strength : num [1:826] 0.343 0.264 0.335 0.542 0.199 ...

$ exclude : NULL

$ net.result :List of 1

..$ : num [1:826, 1] 0.219 0.247 0.281 0.253 0.291 ...

.. ..- attr(\*, "dimnames")=List of 2

.. .. ..$ : chr [1:826] "1" "2" "3" "4" ...

.. .. ..$ : NULL

$ weights :List of 1

..$ :List of 2

.. ..$ : num [1:9, 1] 1.747 -3.991 -2.181 -0.617 1.987 ...

.. ..$ : num [1:2, 1] 0.67 -0.678

$ result.matrix : num [1:14, 1] 5.55 9.01e-03 1.72e+03 1.75 -3.99 ...

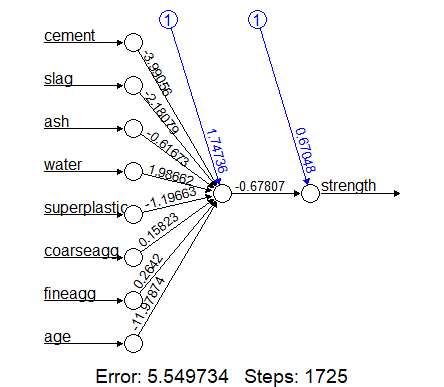
..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:14] "error" "reached.threshold" "steps" "Intercept.to.1layhid1" ...

.. ..$ : NULL

- attr(\*, "class")= chr "nn"

> plot(model\_con)



> pred\_model\_con <-compute(model\_con,tst\_c)

> predict\_con <- pred\_model\_con$net.result

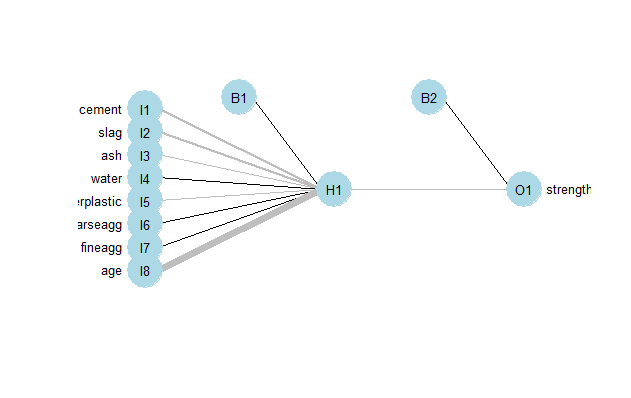
> cor(predict\_con,tst\_c$strength)

[,1]

[1,] 0.8219754

**SSE is more with less acc**

> plotnet(model\_con,cex=0.8)



**Improving model performance by including hidden nodes**

> model2\_con <- neuralnet(strength~.,data = trn\_c,hidden = 5)

> str(model2\_con)

List of 14

$ call : language neuralnet(formula = strength ~ ., data = trn\_c, hidden = 5)

$ response : num [1:826, 1] 0.343 0.264 0.335 0.542 0.199 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:826] "1" "2" "3" "4" ...

.. ..$ : chr "strength"

$ covariate : num [1:826, 1:8] 0.0897 0.1527 0.3379 0.3744 0.1205 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:826] "1" "2" "3" "4" ...

.. ..$ : chr [1:8] "cement" "slag" "ash" "water" ...

$ model.list :List of 2

..$ response : chr "strength"

..$ variables: chr [1:8] "cement" "slag" "ash" "water" ...

$ err.fct :function (x, y)

..- attr(\*, "type")= chr "sse"

$ act.fct :function (x)

..- attr(\*, "type")= chr "logistic"

$ linear.output : logi TRUE

$ data :'data.frame': 826 obs. of 9 variables:

..$ cement : num [1:826] 0.0897 0.1527 0.3379 0.3744 0.1205 ...

..$ slag : num [1:826] 0.59 0.117 0 0.317 0.51 ...

..$ ash : num [1:826] 0 0.621 0.478 0 0 ...

..$ water : num [1:826] 0.653 0.292 0.524 0.848 0.571 ...

..$ superplastic: num [1:826] 0 0.335 0.171 0 0.283 ...

..$ coarseagg : num [1:826] 0.497 0.813 0.453 0.381 0.716 ...

..$ fineagg : num [1:826] 0.388 0.507 0.67 0.191 0.258 ...

..$ age : num [1:826] 0.0742 0.0357 0.0742 0.0742 0.0742 ...

..$ strength : num [1:826] 0.343 0.264 0.335 0.542 0.199 ...

$ exclude : NULL

$ net.result :List of 1

..$ : num [1:826, 1] 0.276 0.29 0.389 0.526 0.31 ...

.. ..- attr(\*, "dimnames")=List of 2

.. .. ..$ : chr [1:826] "1" "2" "3" "4" ...

.. .. ..$ : NULL

$ weights :List of 1

..$ :List of 2

.. ..$ : num [1:9, 1:5] -5.856 2.374 2.628 1.409 0.532 ...

.. ..$ : num [1:6, 1] 1.3826 -3.6579 1.3653 0.0791 -2.3792 ...

$ result.matrix : num [1:54, 1] 1.78 9.46e-03 2.54e+04 -5.86 2.37 ...

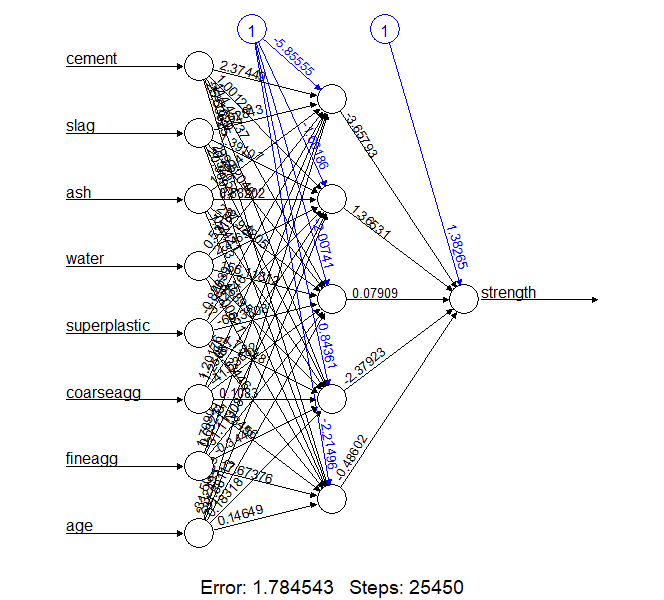
..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:54] "error" "reached.threshold" "steps" "Intercept.to.1layhid1" ...

.. ..$ : NULL

- attr(\*, "class")= chr "nn"

> plot(model2\_con)



> predict\_model2 <- compute(model2\_con,tst\_c)

> predict\_con2 <- predict\_model2$net.result

> cor(predict\_con2,tst\_c$strength)

[,1]

[1,] 0.9383941

**SSE has decreased**

> plotnet(model2\_con,cex=0.8)

