PREDICT THE BURNED AREA OF FOREST FIRES WITH NEURAL NETWORKS

> library(neuralnet)

> library(nnet)

> library(caret)

> library(corrplot)

> library(NeuralNetTools)

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

> View(forest)

> str(forest)

'data.frame': 517 obs. of 31 variables:

$ month : Factor w/ 12 levels "apr","aug","dec",..: 8 11 11 8 8 2 2 2 12 12 ...

$ day : Factor w/ 7 levels "fri","mon","sat",..: 1 6 3 1 4 4 2 2 6 3 ...

$ FFMC : num 86.2 90.6 90.6 91.7 89.3 92.3 92.3 91.5 91 92.5 ...

$ DMC : num 26.2 35.4 43.7 33.3 51.3 ...

$ DC : num 94.3 669.1 686.9 77.5 102.2 ...

$ ISI : num 5.1 6.7 6.7 9 9.6 14.7 8.5 10.7 7 7.1 ...

$ temp : num 8.2 18 14.6 8.3 11.4 22.2 24.1 8 13.1 22.8 ...

$ RH : int 51 33 33 97 99 29 27 86 63 40 ...

$ wind : num 6.7 0.9 1.3 4 1.8 5.4 3.1 2.2 5.4 4 ...

$ rain : num 0 0 0 0.2 0 0 0 0 0 0 ...

$ area : num 0 0 0 0 0 0 0 0 0 0 ...

$ dayfri : int 1 0 0 1 0 0 0 0 0 0 ...

$ daymon : int 0 0 0 0 0 0 1 1 0 0 ...

$ daysat : int 0 0 1 0 0 0 0 0 0 1 ...

$ daysun : int 0 0 0 0 1 1 0 0 0 0 ...

$ daythu : int 0 0 0 0 0 0 0 0 0 0 ...

$ daytue : int 0 1 0 0 0 0 0 0 1 0 ...

$ daywed : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthapr : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthaug : int 0 0 0 0 0 1 1 1 0 0 ...

$ monthdec : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthfeb : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjan : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjul : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjun : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthmar : int 1 0 0 1 1 0 0 0 0 0 ...

$ monthmay : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthnov : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthoct : int 0 1 1 0 0 0 0 0 0 0 ...

$ monthsep : int 0 0 0 0 0 0 0 0 1 1 ...

$ size\_category: Factor w/ 2 levels "large","small": 2 2 2 2 2 2 2 2 2 2 ...

> table(forest$size\_category)

large small

139 378

> colnames(forest)

[1] "month" "day" "FFMC" "DMC"

[5] "DC" "ISI" "temp" "RH"

[9] "wind" "rain" "area" "dayfri"

[13] "daymon" "daysat" "daysun" "daythu"

[17] "daytue" "daywed" "monthapr" "monthaug"

[21] "monthdec" "monthfeb" "monthjan" "monthjul"

[25] "monthjun" "monthmar" "monthmay" "monthnov"

[29] "monthoct" "monthsep" "size\_category"

#Creating dummies

> forest$month=as.integer(factor(forest$month,levels = c("jan","feb","mar","apr","may","jun","jul","aug","sep","oct","nov","dec"),labels = c(1,2,3,4,5,6,7,8,9,10,11,12)))

> forest$day=as.integer(factor(forest$day,levels = c("sun","mon","tue","wed","thu","fri","sat"),labels = c(1,2,3,4,5,6,7)))

> forest$size\_category=as.integer(factor(forest$size\_category,levels = c("large","small"),labels = c(1,0)))

> str(forest)

'data.frame': 517 obs. of 31 variables:

$ month : int 3 10 10 3 3 8 8 8 9 9 ...

$ day : int 6 3 7 6 1 1 2 2 3 7 ...

$ FFMC : num 86.2 90.6 90.6 91.7 89.3 92.3 92.3 91.5 91 92.5 ...

$ DMC : num 26.2 35.4 43.7 33.3 51.3 ...

$ DC : num 94.3 669.1 686.9 77.5 102.2 ...

$ ISI : num 5.1 6.7 6.7 9 9.6 14.7 8.5 10.7 7 7.1 ...

$ temp : num 8.2 18 14.6 8.3 11.4 22.2 24.1 8 13.1 22.8 ...

$ RH : int 51 33 33 97 99 29 27 86 63 40 ...

$ wind : num 6.7 0.9 1.3 4 1.8 5.4 3.1 2.2 5.4 4 ...

$ rain : num 0 0 0 0.2 0 0 0 0 0 0 ...

$ area : num 0 0 0 0 0 0 0 0 0 0 ...

$ dayfri : int 1 0 0 1 0 0 0 0 0 0 ...

$ daymon : int 0 0 0 0 0 0 1 1 0 0 ...

$ daysat : int 0 0 1 0 0 0 0 0 0 1 ...

$ daysun : int 0 0 0 0 1 1 0 0 0 0 ...

$ daythu : int 0 0 0 0 0 0 0 0 0 0 ...

$ daytue : int 0 1 0 0 0 0 0 0 1 0 ...

$ daywed : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthapr : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthaug : int 0 0 0 0 0 1 1 1 0 0 ...

$ monthdec : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthfeb : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjan : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjul : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjun : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthmar : int 1 0 0 1 1 0 0 0 0 0 ...

$ monthmay : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthnov : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthoct : int 0 1 1 0 0 0 0 0 0 0 ...

$ monthsep : int 0 0 0 0 0 0 0 0 1 1 ...

$ size\_category: int 2 2 2 2 2 2 2 2 2 2 ...

> normalize <- function(x){

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

}

> norm\_foresr <- as.data.frame(lapply(forest,normalize))

> View(norm\_foresr)

> attach(norm\_foresr)

The following objects are masked from norm\_foresr (pos = 3):

area, day, dayfri, daymon, daysat, daysun, daythu, daytue, daywed,

DC, DMC, FFMC, ISI, month, monthapr, monthaug, monthdec, monthfeb,

monthjan, monthjul, monthjun, monthmar, monthmay, monthnov,

monthoct, monthsep, rain, RH, size\_category, temp, wind

#Splitting of data to train and test

> train <- sample(2,nrow(norm\_foresr),prob = c(0.7,0.3),replace = T)

> trn <- norm\_foresr[train==1,]

> tst <- norm\_foresr[train==2,]

#Model building

> model <- neuralnet(area~.,data = trn,rep=5)

> str(model)

List of 14

$ call : language neuralnet(formula = area ~ ., data = trn, rep = 5)

$ response : num [1:356, 1] 0 0 0 0 0 0 0 0 0 0 ...

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

.. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. ..$ : chr "area"

$ covariate : num [1:356, 1:30] 0.818 0.182 0.182 0.636 0.727 ...

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

.. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. ..$ : chr [1:30] "month" "day" "FFMC" "DMC" ...

$ model.list :List of 2

..$ response : chr "area"

..$ variables: chr [1:30] "month" "day" "FFMC" "DMC" ...

$ err.fct :function (x, y)

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

$ act.fct :function (x)

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

$ linear.output : logi TRUE

$ data :'data.frame': 356 obs. of 31 variables:

..$ month : num [1:356] 0.818 0.182 0.182 0.636 0.727 ...

..$ day : num [1:356] 0.333 0.833 0 0 0.333 ...

..$ FFMC : num [1:356] 0.928 0.942 0.911 0.95 0.933 ...

..$ DMC : num [1:356] 0.118 0.111 0.173 0.29 0.442 ...

..$ DC : num [1:356] 0.7754 0.0816 0.1106 0.563 0.803 ...

..$ ISI : num [1:356] 0.119 0.16 0.171 0.262 0.125 ...

..$ temp : num [1:356] 0.508 0.196 0.296 0.643 0.35 ...

..$ RH : num [1:356] 0.212 0.965 0.988 0.165 0.565 ...

..$ wind : num [1:356] 0.0556 0.4 0.1556 0.5556 0.5556 ...

..$ rain : num [1:356] 0 0.0312 0 0 0 ...

..$ area : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ dayfri : num [1:356] 0 1 0 0 0 0 0 0 1 0 ...

..$ daymon : num [1:356] 0 0 0 0 0 0 1 0 0 0 ...

..$ daysat : num [1:356] 0 0 0 0 0 1 0 0 0 0 ...

..$ daysun : num [1:356] 0 0 1 1 0 0 0 0 0 0 ...

..$ daythu : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ daytue : num [1:356] 1 0 0 0 1 0 0 0 0 0 ...

..$ daywed : num [1:356] 0 0 0 0 0 0 0 1 0 1 ...

..$ monthapr : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthaug : num [1:356] 0 0 0 1 0 0 0 0 0 0 ...

..$ monthdec : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthfeb : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthjan : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthjul : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthjun : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthmar : num [1:356] 0 1 1 0 0 0 0 0 0 1 ...

..$ monthmay : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthnov : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthoct : num [1:356] 1 0 0 0 0 0 0 0 0 0 ...

..$ monthsep : num [1:356] 0 0 0 0 1 1 1 1 1 0 ...

..$ size\_category: num [1:356] 1 1 1 1 1 1 1 1 1 1 ...

$ exclude : NULL

$ net.result :List of 5

..$ : num [1:356, 1] 0.00627 0.00627 0.00627 0.00627 0.00627 ...

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

.. .. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. .. ..$ : NULL

..$ : num [1:356, 1] 0.01407 -0.01363 -0.01823 0.01298 0.00944 ...

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

.. .. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. .. ..$ : NULL

..$ : num [1:356, 1] 0.0065 0.0065 0.0065 0.0065 0.0065 ...

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

.. .. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. .. ..$ : NULL

..$ : num [1:356, 1] -0.00391 0.01054 0.01506 -0.01018 0.00766 ...

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

.. .. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. .. ..$ : NULL

..$ : num [1:356, 1] 0.00604 0.00604 0.00604 0.00612 0.00604 ...

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

.. .. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. .. ..$ : NULL

$ weights :List of 5

..$ :List of 2

.. ..$ : num [1:31, 1] -1.251 -0.372 -1.016 -0.701 1.334 ...

.. ..$ : num [1:2, 1] 0.00627 1.62702

..$ :List of 2

.. ..$ : num [1:31, 1] -2.202 0.94 -0.944 0.039 0.625 ...

.. ..$ : num [1:2, 1] 0.0293 -0.109

..$ :List of 2

.. ..$ : num [1:31, 1] -0.11 1.34 1.61 2.26 -5.73 ...

.. ..$ : num [1:2, 1] 0.697 -0.691

..$ :List of 2

.. ..$ : num [1:31, 1] -0.1155 0.8079 0.0978 2.6836 -0.1915 ...

.. ..$ : num [1:2, 1] -0.412 0.454

..$ :List of 2

.. ..$ : num [1:31, 1] -0.972 -2.324 0.39 -2.582 1.738 ...

.. ..$ : num [1:2, 1] 0.00604 1.56781

$ generalized.weights:List of 5

..$ : num [1:356, 1:30] -1.37e-08 -4.64e-10 -1.57e-10 -9.11e-07 -3.98e-09 ...

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

.. .. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. .. ..$ : NULL

..$ : num [1:356, 1:30] -0.885 1.769 1.357 -1.015 -1.628 ...

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

.. .. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. .. ..$ : NULL

..$ : num [1:356, 1:30] 0 0 0 0 0 0 0 0 0 0 ...

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

.. .. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. .. ..$ : NULL

..$ : num [1:356, 1:30] -8.55 2.29 1.39 -3.65 3.4 ...

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

.. .. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. .. ..$ : NULL

..$ : num [1:356, 1:30] -2.15e-04 -1.73e-07 -9.76e-07 -3.16e-02 -1.18e-05 ...

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

.. .. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. .. ..$ : NULL

$ startweights :List of 5

..$ :List of 2

.. ..$ : num [1:31, 1] -0.326 0.552 -0.675 0.214 0.311 ...

.. ..$ : num [1:2, 1] 0.279 1.008

..$ :List of 2

.. ..$ : num [1:31, 1] -2.073 1.19 -0.724 0.168 0.92 ...

.. ..$ : num [1:2, 1] 0.423 0.387

..$ :List of 2

.. ..$ : num [1:31, 1] -0.6878 0.1489 -0.0576 -0.0748 1.5099 ...

.. ..$ : num [1:2, 1] 1.365 0.497

..$ :List of 2

.. ..$ : num [1:31, 1] -0.814 0.268 -0.592 2.133 1.173 ...

.. ..$ : num [1:2, 1] -0.783 0.245

..$ :List of 2

.. ..$ : num [1:31, 1] -0.144 -1.609 0.952 -1.819 1.784 ...

.. ..$ : num [1:2, 1] -1.587 -0.927

$ result.matrix : num [1:36, 1:5] 0.11198 0.00973 149 -1.25081 -0.37231 ...

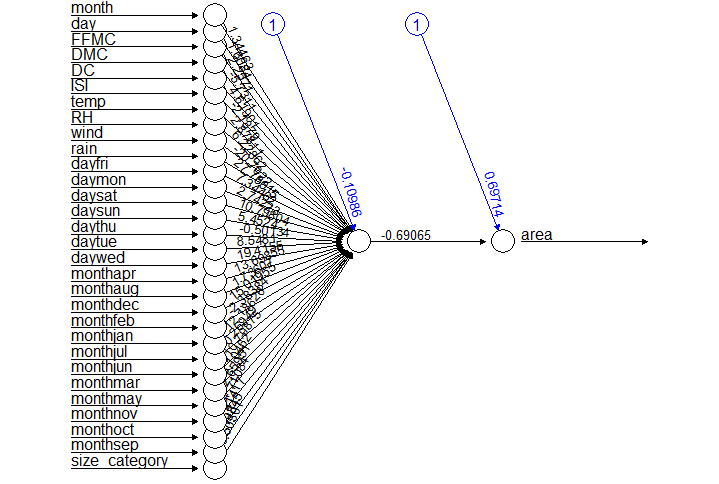
..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:36] "error" "reached.threshold" "steps" "Intercept.to.1layhid1" ...

.. ..$ : NULL

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

> plot(model,"best")



> predict\_model <- compute(model,tst)

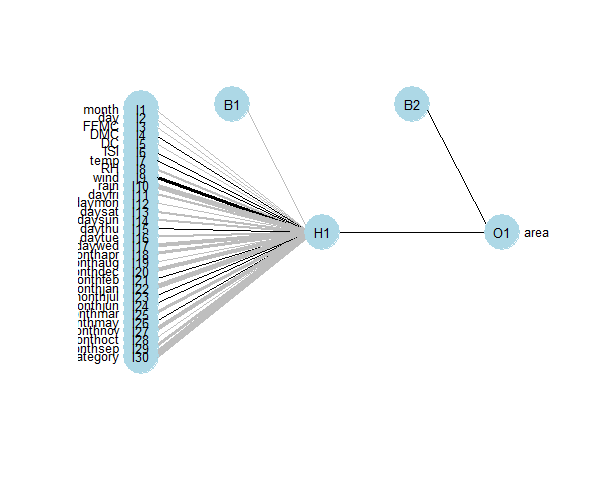
> pred <- predict\_model$net.result

> cor(pred,tst$area)

[,1]

[1,] -0.00210729

#Accuracy is very poor

> plotnet(model,cex=0.8)

#Improving performance of the model by including hidden nodes

> model2 <- neuralnet(area~.,hidden = 7,data = trn,)

#Hidden chosen from bagging method

> str(model2)

List of 14

$ call : language neuralnet(formula = area ~ ., data = trn, hidden = 7)

$ response : num [1:356, 1] 0 0 0 0 0 0 0 0 0 0 ...

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

.. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. ..$ : chr "area"

$ covariate : num [1:356, 1:30] 0.818 0.182 0.182 0.636 0.727 ...

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

.. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. ..$ : chr [1:30] "month" "day" "FFMC" "DMC" ...

$ model.list :List of 2

..$ response : chr "area"

..$ variables: chr [1:30] "month" "day" "FFMC" "DMC" ...

$ err.fct :function (x, y)

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

$ act.fct :function (x)

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

$ linear.output : logi TRUE

$ data :'data.frame': 356 obs. of 31 variables:

..$ month : num [1:356] 0.818 0.182 0.182 0.636 0.727 ...

..$ day : num [1:356] 0.333 0.833 0 0 0.333 ...

..$ FFMC : num [1:356] 0.928 0.942 0.911 0.95 0.933 ...

..$ DMC : num [1:356] 0.118 0.111 0.173 0.29 0.442 ...

..$ DC : num [1:356] 0.7754 0.0816 0.1106 0.563 0.803 ...

..$ ISI : num [1:356] 0.119 0.16 0.171 0.262 0.125 ...

..$ temp : num [1:356] 0.508 0.196 0.296 0.643 0.35 ...

..$ RH : num [1:356] 0.212 0.965 0.988 0.165 0.565 ...

..$ wind : num [1:356] 0.0556 0.4 0.1556 0.5556 0.5556 ...

..$ rain : num [1:356] 0 0.0312 0 0 0 ...

..$ area : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ dayfri : num [1:356] 0 1 0 0 0 0 0 0 1 0 ...

..$ daymon : num [1:356] 0 0 0 0 0 0 1 0 0 0 ...

..$ daysat : num [1:356] 0 0 0 0 0 1 0 0 0 0 ...

..$ daysun : num [1:356] 0 0 1 1 0 0 0 0 0 0 ...

..$ daythu : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ daytue : num [1:356] 1 0 0 0 1 0 0 0 0 0 ...

..$ daywed : num [1:356] 0 0 0 0 0 0 0 1 0 1 ...

..$ monthapr : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthaug : num [1:356] 0 0 0 1 0 0 0 0 0 0 ...

..$ monthdec : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthfeb : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthjan : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthjul : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthjun : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthmar : num [1:356] 0 1 1 0 0 0 0 0 0 1 ...

..$ monthmay : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthnov : num [1:356] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthoct : num [1:356] 1 0 0 0 0 0 0 0 0 0 ...

..$ monthsep : num [1:356] 0 0 0 0 1 1 1 1 1 0 ...

..$ size\_category: num [1:356] 1 1 1 1 1 1 1 1 1 1 ...

$ exclude : NULL

$ net.result :List of 1

..$ : num [1:356, 1] -0.00455 0.0017 -0.00968 -0.00305 0.01272 ...

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

.. .. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. .. ..$ : NULL

$ weights :List of 1

..$ :List of 2

.. ..$ : num [1:31, 1:7] -0.3973 0.2566 0.0429 0.9899 -0.1322 ...

.. ..$ : num [1:8, 1] -0.902 0.944 -0.179 -0.478 0.763 ...

$ generalized.weights:List of 1

..$ : num [1:356, 1:30] 62.7 -153.9 32.2 28.4 -15.2 ...

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

.. .. ..$ : chr [1:356] "2" "4" "5" "6" ...

.. .. ..$ : NULL

$ startweights :List of 1

..$ :List of 2

.. ..$ : num [1:31, 1:7] -0.5022 0.1315 -0.0789 0.8868 0.117 ...

.. ..$ : num [1:8, 1] -0.9673 0.8794 -0.2536 -1.5179 -0.0239 ...

$ result.matrix : num [1:228, 1] 0.06467 0.00931 665 -0.39732 0.25665 ...

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

.. ..$ : chr [1:228] "error" "reached.threshold" "steps" "Intercept.to.1layhid1" ...

.. ..$ : NULL

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

> pred2model <- compute(model2,tst)

> pred2 <- pred2model$net.result

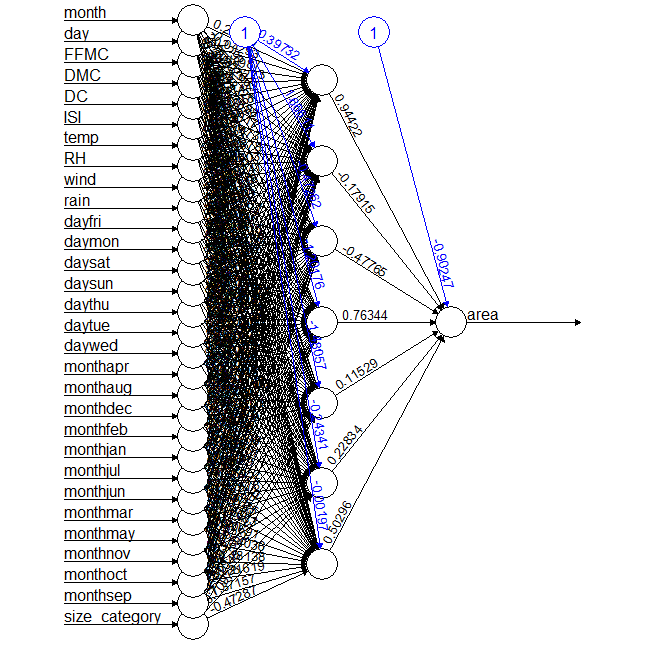
> cor(pred2,tst$area)

[,1]

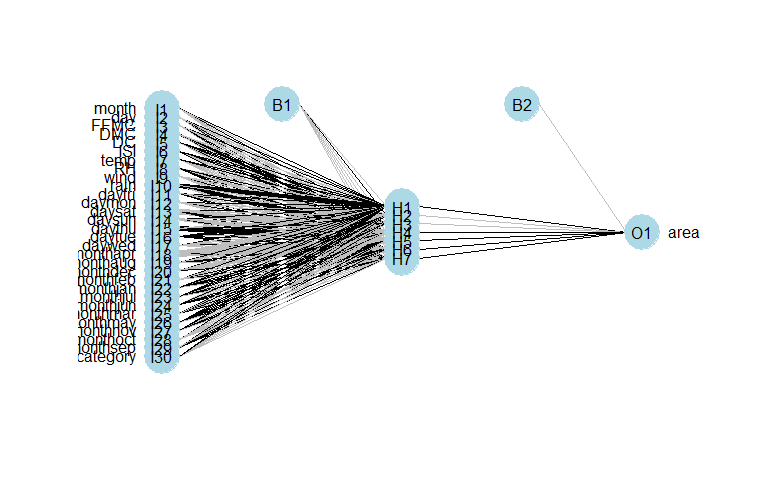
[1,] 0.04630228

#SSE is decreased

> plot(model2)



> plotnet(model2)



**#**Bagging

> acc <- c()

>for(i in seq(1,10,1)){

model\_bag <- neuralnet(area~.,hidden = i,data = trn)

pred\_bag <- compute(model\_bag,tst)

acc <- c(acc,cor(pred\_bag$net.result,tst$area))

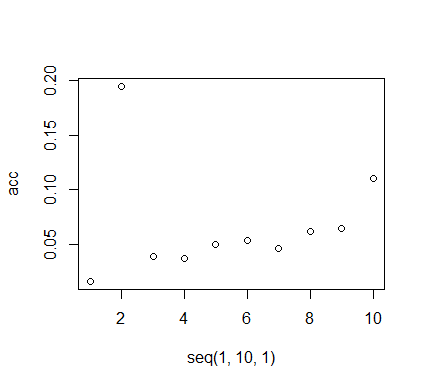
}

> acc

[1] 0.01636077 0.19447745 0.03911585 0.03765625 0.05007557 0.05374143 0.04630228

[8] 0.06170272 0.06438313 0.11020099

> plot(seq(1,10,1),acc)



#Accuracy is Only good at 2, i.e. 0.19447745