NN-Regression–Yacht-Hydrodynamics.R

patriciamaya

2020-11-30

```
#ANN
#Our regression ANN will use the Yacht Hydrodynamics data set from UCI's Machine Learning Repository.
#This data set contains data contains results from 308 full-scale experiments performed at the Delft
#Ship Hydromechanics Laboratory where they test 22 different hull forms. Their experiment tested the
#effect of variations in the hull geometry and the ship's Froude number on the craft's residuary
#resistance per unit weight of displacement.

library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.2
```

```
## -- Attaching packages --------------------------------------------------------------- tidyverse 1.3.0 -
```

```
## v ggplot2 3.3.1     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## Warning: package 'forcats' was built under R version 4.0.2
```

```
## -- Conflicts ------------------------------------------------------------------ tidyverse_conflicts() -
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(neuralnet)
```

```
## Warning: package 'neuralnet' was built under R version 4.0.2
```

```
##
## Attaching package: 'neuralnet'
```

```
## The following object is masked from 'package:dplyr':
##
##     compute
```

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.0.2
```

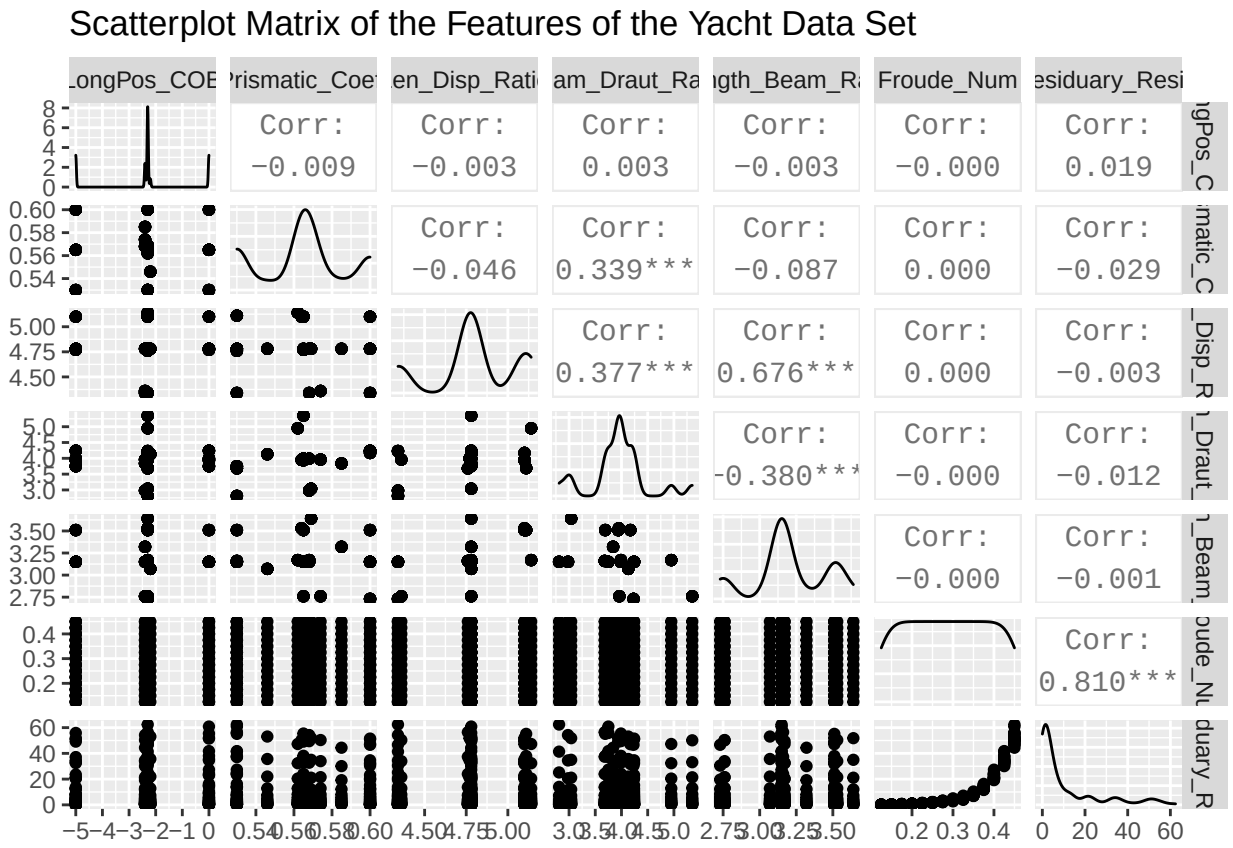
```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
url <- 'http://archive.ics.uci.edu/ml/machine-learning-databases/00243/yacht_hydrodynamics.data'

Yacht_Data <- read_table(file = url, col_names = c('LongPos_COB', 'Prismatic_Coeff','Len_Disp_Ratio', '
```

```
## Parsed with column specification:
```

```
## cols(
##   LongPos_COB = col_double(),
##   Prismatic_Coeff = col_double(),
##   Len_Disp_Ratio = col_double(),
##   Beam_Draut_Ratio = col_double(),
##   Length_Beam_Ratio = col_double(),
##   Froude_Num = col_double(),
##   Residuary_Resist = col_double()
## )
```

```
ggpairs(Yacht_Data, title = "Scatterplot Matrix of the Features of the Yacht Data Set")
```

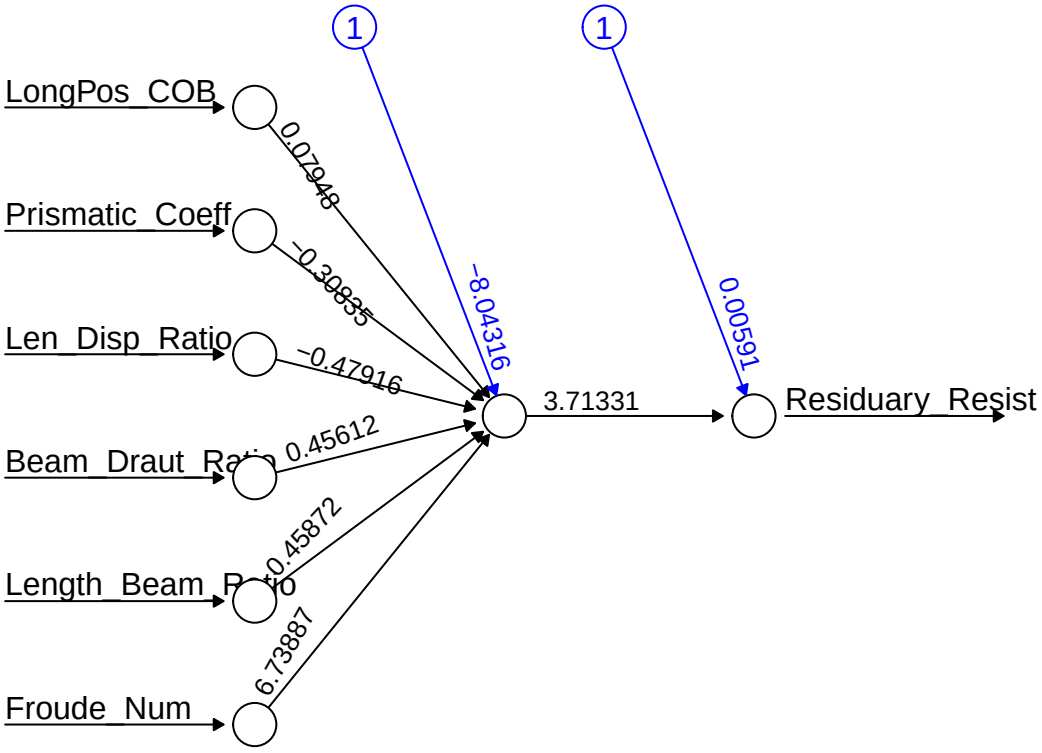



```
# Scale the Data
scale01 <- function(x){
  (x - min(x)) / (max(x) - min(x))
}

Yacht_Data <- Yacht_Data %>%
  mutate_all(scale01)

# Split into test and train sets
set.seed(12345)
Yacht_Data_Train <- sample_frac(tbl = Yacht_Data, replace = FALSE, size = 0.80)
Yacht_Data_Test <- anti_join(Yacht_Data, Yacht_Data_Train)
```

```
## Joining, by = c("LongPos_COB", "Prismatic_Coeff", "Len_Disp_Ratio", "Beam_Draut_Ratio", "Length_Beam_
```

```
set.seed(12321)
Yacht_NN1 <- neuralnet(Residuary_Resist ~ LongPos_COB + Prismatic_Coeff +
```

```
                        Len_Disp_Ratio + Beam_Draut_Ratio + Length_Beam_Ratio +
                        Froude_Num, data = Yacht_Data_Train)

plot(Yacht_NN1, rep = 'best')
```



Error: 0.036462 Steps: 1499

```
#manually compute the SSE you can use the following:
NN1_Train_SSE <- sum((Yacht_NN1$net.result - Yacht_Data_Train[, 7])^2)/2
paste("SSE: ", round(NN1_Train_SSE, 4))
```

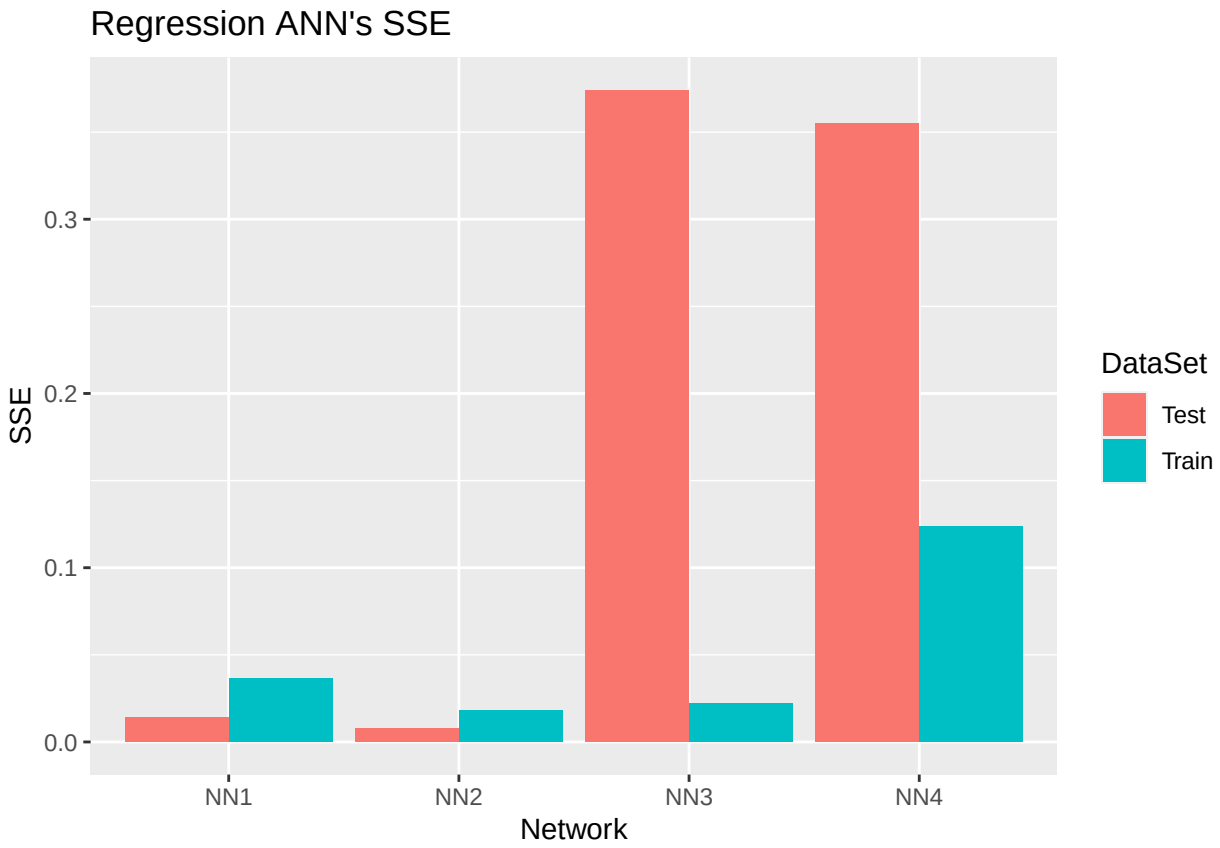
```
## [1] "SSE:  0.0365"
```

```
Test_NN1_Output <- compute(Yacht_NN1, Yacht_Data_Test[, 1:6])$net.result
NN1_Test_SSE <- sum((Test_NN1_Output - Yacht_Data_Test[, 7])^2)/2
NN1_Test_SSE
```

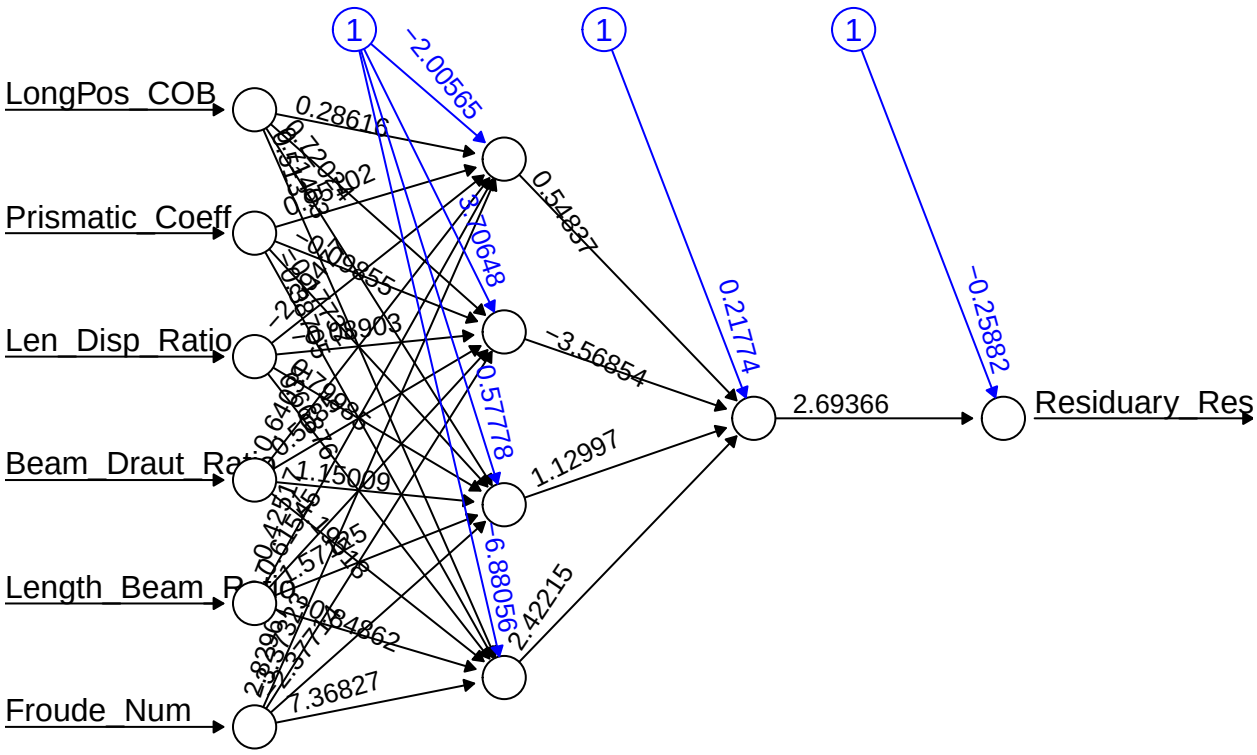
```
## [1] 0.01387174
```

```
# *** Regression Hyperparameters
# 2-Hidden Layers, Layer-1 4-neurons, Layer-2, 1-neuron, logistic activation
# function
set.seed(12321)
Yacht_NN2 <- neuralnet(Residuary_Resist ~ LongPos_COB + Prismatic_Coeff + Len_Disp_Ratio + Beam_Draut_Ra
                       data = Yacht_Data_Train,
                       hidden = c(4, 1),
                       act.fct = "logistic")

## Training Error
NN2_Train_SSE <- sum((Yacht_NN2$net.result - Yacht_Data_Train[, 7])^2)/2
```

```r
## Test Error
Test_NN2_Output <- compute(Yacht_NN2, Yacht_Data_Test[, 1:6])$net.result
NN2_Test_SSE <- sum((Test_NN2_Output - Yacht_Data_Test[, 7])^2)/2

# Rescale for tanh activation function
scale11 <- function(x) {
  (2 * ((x - min(x))/(max(x) - min(x)))) - 1
}
Yacht_Data_Train <- Yacht_Data_Train %>% mutate_all(scale11)
Yacht_Data_Test <- Yacht_Data_Test %>% mutate_all(scale11)

# 2-Hidden Layers, Layer-1 4-neurons, Layer-2, 1-neuron, tanh activation
# function
set.seed(12321)
Yacht_NN3 <- neuralnet(Residuary_Resist ~ LongPos_COB + Prismatic_Coeff + Len_Disp_Ratio + Beam_Draut_Ra
                       data = Yacht_Data_Train,
                       hidden = c(4, 1),
                       act.fct = "tanh")

## Training Error
NN3_Train_SSE <- sum((Yacht_NN3$net.result - Yacht_Data_Train[, 7])^2)/2

## Test Error
Test_NN3_Output <- compute(Yacht_NN3, Yacht_Data_Test[, 1:6])$net.result
NN3_Test_SSE <- sum((Test_NN3_Output - Yacht_Data_Test[, 7])^2)/2

# 1-Hidden Layer, 1-neuron, tanh activation function
set.seed(12321)
Yacht_NN4 <- neuralnet(Residuary_Resist ~ LongPos_COB + Prismatic_Coeff + Len_Disp_Ratio + Beam_Draut_Ra
                       data = Yacht_Data_Train,
                       act.fct = "tanh")

## Training Error
NN4_Train_SSE <- sum((Yacht_NN4$net.result - Yacht_Data_Train[, 7])^2)/2

## Test Error
Test_NN4_Output <- compute(Yacht_NN4, Yacht_Data_Test[, 1:6])$net.result
NN4_Test_SSE <- sum((Test_NN4_Output - Yacht_Data_Test[, 7])^2)/2

# Bar plot of results
Regression_NN_Errors <- tibble(Network = rep(c("NN1", "NN2", "NN3", "NN4"), each = 2),
                               DataSet = rep(c("Train", "Test"), time = 4),
                               SSE = c(NN1_Train_SSE, NN1_Test_SSE,
                                       NN2_Train_SSE, NN2_Test_SSE,
                                       NN3_Train_SSE, NN3_Test_SSE,
                                       NN4_Train_SSE, NN4_Test_SSE))

Regression_NN_Errors %>%
  ggplot(aes(Network, SSE, fill = DataSet)) +
  geom_col(position = "dodge") +
  ggtitle("Regression ANN's SSE")
```




```
#As evident from the plot, we see that the best regression ANN we found was Yacht_NN2 with a training a
#test SSE of 0.0188 and 0.0057. We make this determination by the value of the training and test SSEs o

plot(Yacht_NN2, rep = "best")
```



Error: 0.017855 Steps: 1013