Redes Neurais e Regressão Polinomial

Um estudo de caso

Luiz Fernando Palin Droubi[*]
Carlos Augusto Zilli[†]
Norberto Hochheim[‡]

10/10/2018

1 INTRODUÇÃO

2 DESENVOLVIMENTO E FUNDAMENTAÇÃO

```
## Creating index variable

# Read the Data
data = read.csv("cereals.csv", header=T)

# Random sampling
samplesize = 0.60 * nrow(data)
set.seed(80)
index = sample( seq_len ( nrow ( data ) ), size = samplesize )

# Create training and test set
datatrain = data[ index, ]
datatest = data[ -index, ]
```

```
## Scale data for neural network

max = apply(data , 2 , max)
min = apply(data, 2 , min)
scaled = as.data.frame(scale(data, center = min, scale = max - min))
```

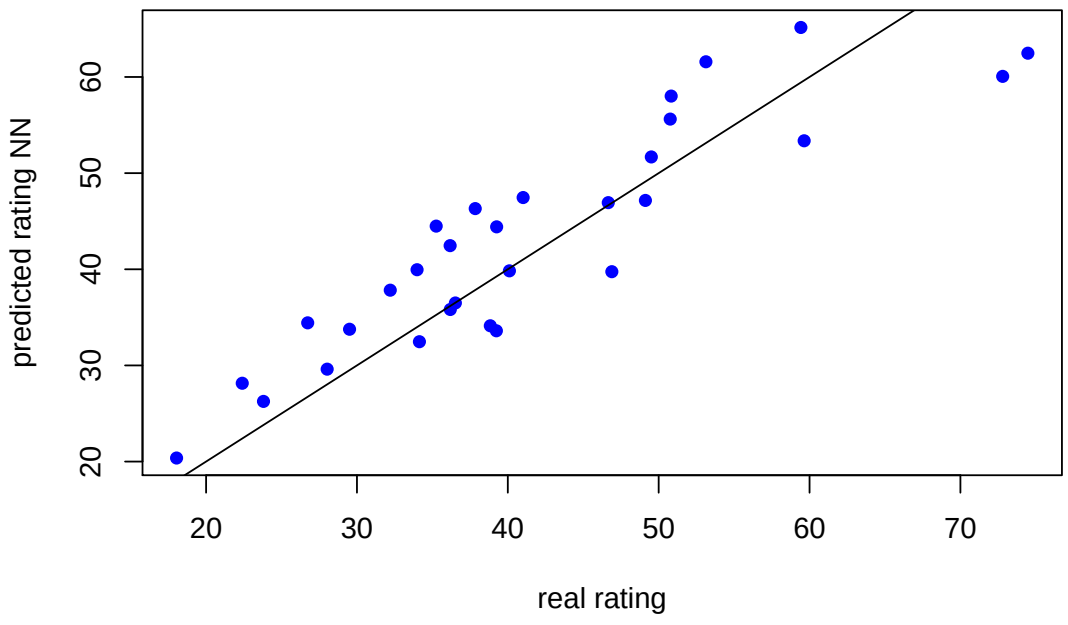
```
## Fit neural network

# load library
library(neuralnet)

# creating training and test set
trainNN = scaled[index , ]
testNN = scaled[-index , ]
```

[*]UFSC, luiz.droubi@planejamento.gov.br
[†]UFSC, carloszilli@hotmail.com
[‡]UFSC, hochheim@gmail.com

```
# fit neural network
set.seed(2)
NN = neuralnet(rating ~ calories + protein + fat + sodium + fiber, trainNN,
               hidden = 3 , linear.output = T )

# plot neural network
plot(NN)
```

2.1 Estimativas

```
## Prediction using neural network

predict_testNN = compute(NN, testNN[,c(1:5)])
predict_testNN = (predict_testNN$net.result * (max(data$rating) - min(data$rating

plot(datatest$rating, predict_testNN, col='blue', pch=16, ylab = "predicted ratin

abline(0,1)
```




```
# Calculate Root Mean Square Error (RMSE)
RMSE.NN = (sum((datatest$rating - predict_testNN)^2) / nrow(datatest)) ^ 0.5
```

2.2 Validação Cruzada

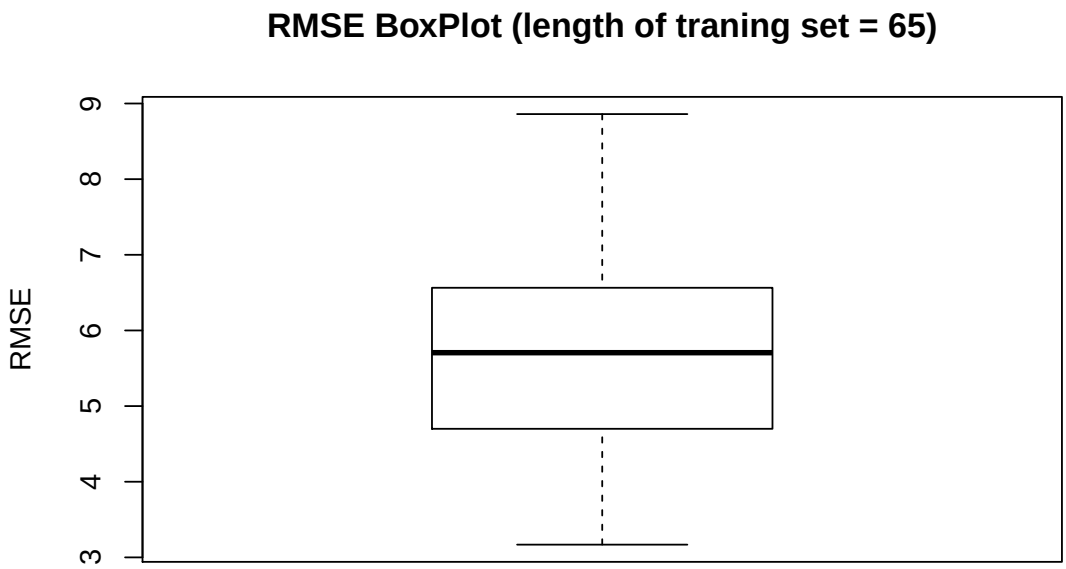
```
## Cross validation of neural network model

# Load libraries
library(boot)
```

```
##
## Attaching package: 'boot'
```

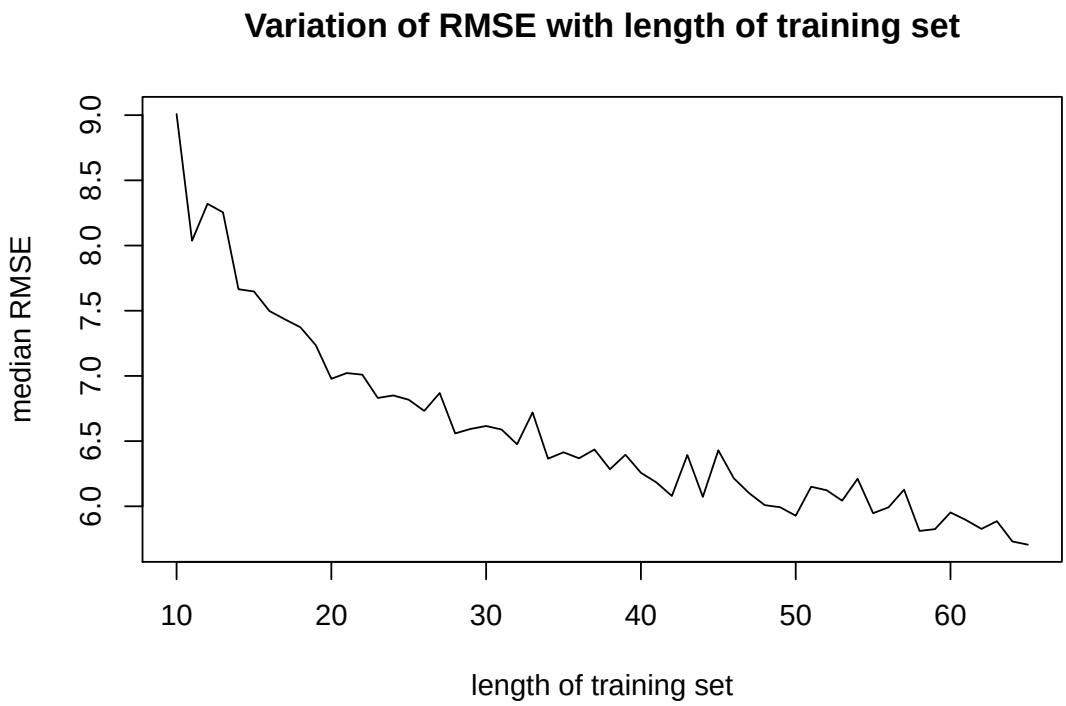
```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

```
## The following object is masked from 'package:car':
##
##     logit
```

```
library(plyr)

# Initialize variables
set.seed(50)
k = 100
RMSE.NN = NULL

List = list( )

# Fit neural network model within nested for loop
for(j in 10:65){
    for (i in 1:k) {
        index = sample(1:nrow(data),j )

        trainNN = scaled[index,]
        testNN = scaled[-index,]
        datatest = data[-index,]

        NN = neuralnet(rating ~ calories + protein + fat + sodium + fiber, trainN
        predict_testNN = compute(NN,testNN[,c(1:5)])
        predict_testNN = (predict_testNN$net.result*(max(data$rating)-min(data$ra

        RMSE.NN [i]<- (sum((datatest$rating - predict_testNN)^2)/nrow(datatest))^
    }
    List[[j]] = RMSE.NN
}

Matrix.RMSE = do.call(cbind, List)

## Prepare boxplot
boxplot(Matrix.RMSE[,56], ylab = "RMSE", main = "RMSE BoxPlot (length of traning
```




```
## Variation of median RMSE
library(matrixStats)

med = colMedians(Matrix.RMSE)

X = seq(10,65)

plot (med~X, type = "l", xlab = "length of training set", ylab = "median RMSE", m
```



3 CONCLUSÕES E RECOMENDAÇÕES

4 REFERÊNCIAS