Redes Neurais e Regressão Polinomial

Um estudo de caso

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

22/10/2018

```
## Creating index variable

# Read the Data
data = centro_2015@data[complete.cases(centro_2015@data),]
data$padrao <- as.numeric(data$padrao)

# 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, ]
```

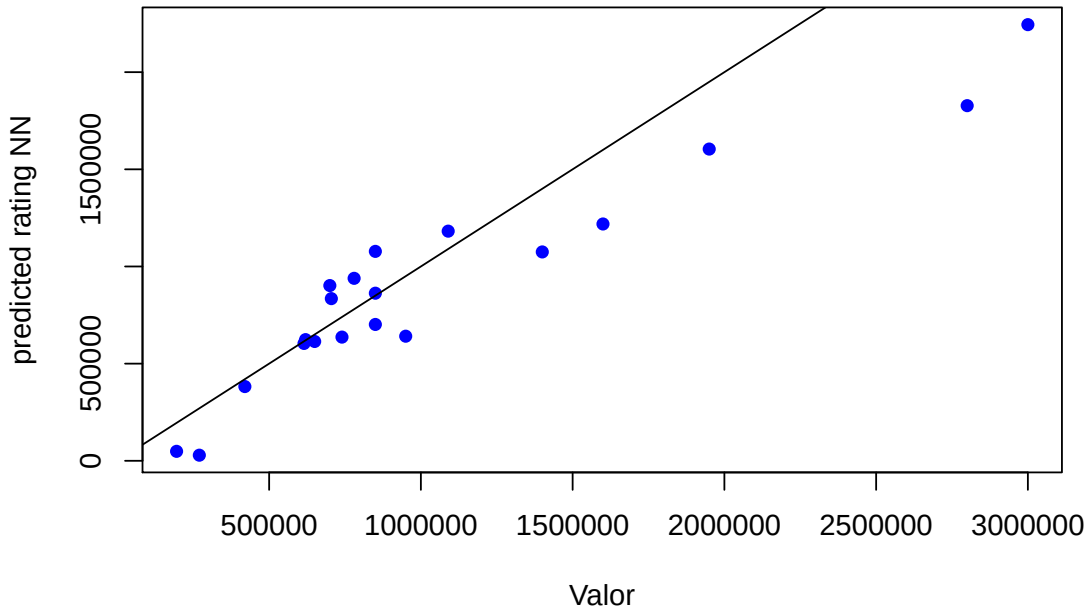
```
## Scale data for neural network
max = apply(data, 2, max)
min = apply(data, 2, min)
scaled = as.data.frame(scale(data, center = min, scale = max - min))
```

```
## Fit neural network

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

# fit neural network
set.seed(2)
NN = neuralnet(valor ~ area_total + quartos + suites + garagens +
                 dist_b_mar + padrao,
               data = trainNN, hidden = 1, linear.output = T )

# plot neural network
plot(NN)
```

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

0.1 Estimativas

```
## Prediction using neural network

predict_testNN = compute(NN, testNN[,2:7])
predict_testNN = (predict_testNN$net.result * (max(data$valor) -
                                                 min(data$valor))) +
  min(data$valor)

plot(datatest$valor, predict_testNN, col='blue', pch=16,
     ylab = "predicted rating NN", xlab = "Valor")

abline(0,1)
```

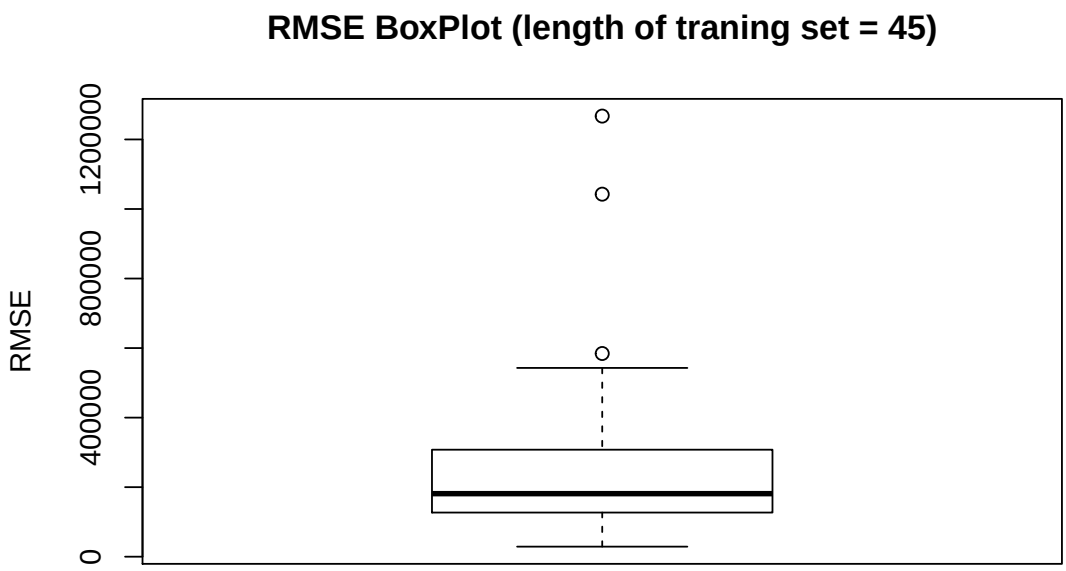



```
# Calculate Root Mean Square Error (RMSE)
RMSE.NN = (sum((datatest$valor -
                  predict_testNN)^2) / nrow(datatest)) ^ 0.5
```

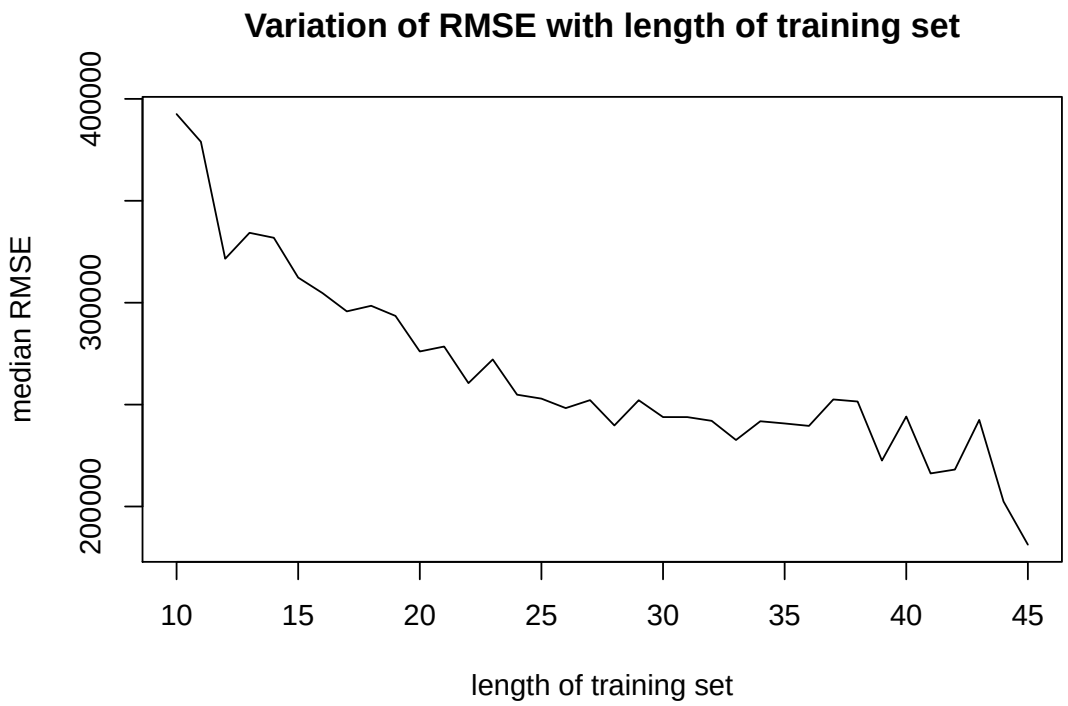
0.2 Validação Cruzada

```
## Cross validation of neural network model

# Load libraries
library(boot)
library(plyr)
```

```r
# Initialize variables
set.seed(50)
k = 100
RMSE.NN = NULL
n <- nrow(data)

List = list()

# Fit neural network model within nested for loop
for(j in seq(0.2, 0.9, 0.02)){
    for (i in 1:k) {
        index = sample(1:n,j*n )

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

        NN = neuralnet(valor ~ area_total + quartos + suites +
                         garagens + dist_b_mar + padrao,
                       trainNN, hidden = 1, linear.output= T)
        predict_testNN = compute(NN,testNN[,c(2:7)])
        predict_testNN = (predict_testNN$net.result*
                            (max(data$valor) - min(data$valor))) +
                          min(data$valor)

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

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

## Prepare boxplot
boxplot(Matrix.RMSE[,36],
        ylab = "RMSE",
        main = "RMSE BoxPlot (length of traning set = 45)")
```



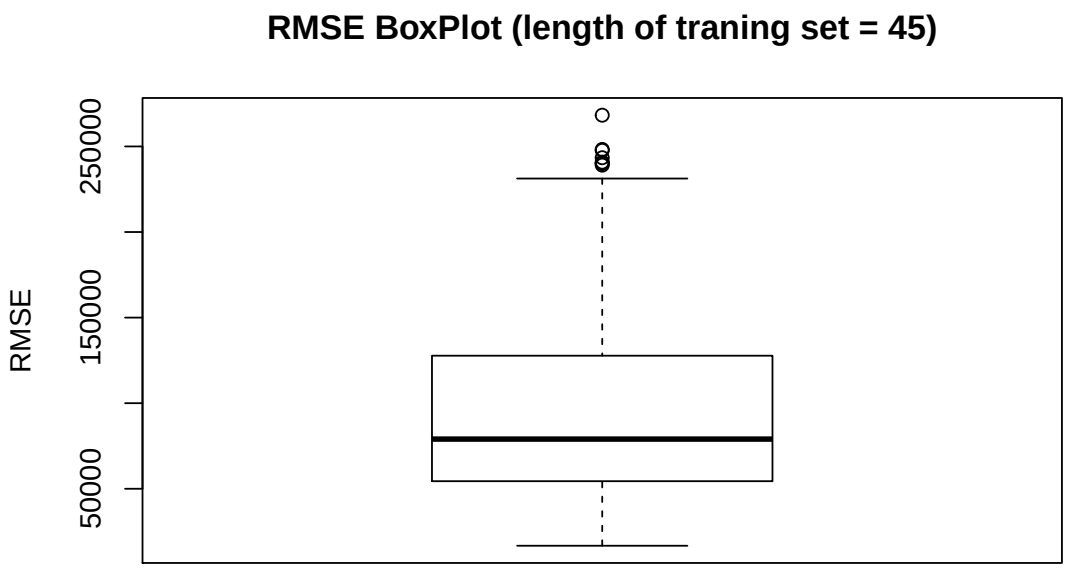

```
## Variation of median RMSE
library(matrixStats)

med = colMedians(Matrix.RMSE)

X = seq(0.2, 0.9, 0.02)*n

plot (med~X, type = "l",
      xlab = "length of training set",
      ylab = "median RMSE",
      main = "Variation of RMSE with length of training set")
```



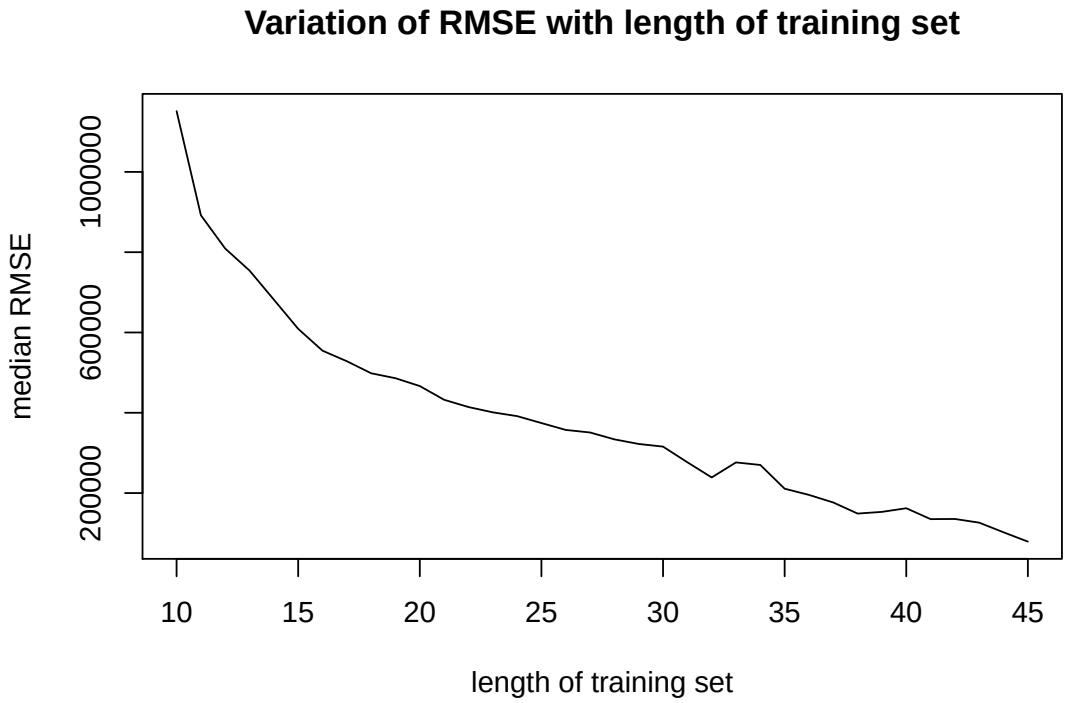
1 REGRESSÃO POLINOMIAL

```
data <- cbind(data[, -1], data[,1])

# Initialize variables
set.seed(50)
k = 100
RMSE.PR = NULL
n <- nrow(data)

List = list()

# Fit PR model within nested for loop
for(j in seq(0.2, 0.9, 0.02)){
  for (i in 1:k) {
    index = sample(1:n,j*n )
    trainPR = data[index,]
    testPR = data[-index,]
    polyFit.out <- polyFit(trainPR, deg = 2, maxInteractDeg = 2,
                           use = "lm", pcaMethod = "prcomp")
    predict_testPR = predict(polyFit.out, testPR[,c(1:6)])
    RMSE.PR[i]<- sqrt(sum((testPR$valor - predict_testPR)^2)/
                        nrow(trainPR))
    }
```

```
  List[[j*n]] = RMSE.PR
}


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

## Prepare boxplot
boxplot(Matrix.RMSE[,36],
        ylab = "RMSE",
        main = "RMSE BoxPlot (length of traning set = 45)")
```




```
## Variation of median RMSE
med = colMedians(Matrix.RMSE)

X = seq(0.2, 0.9, 0.02)*n

plot (med~X, type = "l",
      xlab = "length of training set",
      ylab = "median RMSE",
      main = "Variation of RMSE with length of training set")
```



2 CONCLUSÕES E RECOMENDAÇÕES

3 REFERÊNCIAS